Sorin Larionescu

Sisteme automate

Vol. 1

Universitatea Tehnică de Construcţii Bucureşti

v.8.0/2007

CUPRINS

1. PREFAȚĂ 9

2. INTRODUCERE ÎN TEHNICA SISTEMELOR AUTOMATE 10

2.1. ISTORIA SISTEMELOR AUTOMATE 11
2.2. SISTEM CU CONDUCERE MANUALĂ 12
2.2.1. Schema tehnologică. 12
2.2.2. Schema bloc 13
2.2.3. Algoritmul de conducere manuală. 13
2.3. SEMNALE 14
2.3.1. Semnale discrete logice 14
2.3.2. Semnale continue 15
2.3.3. Semnale eşantionate 15
2.4. SISTEM CU AUTOMAT CU CONTACTE ŞI RELEE. 15
2.4.1. Schema tehnologică cu echipamentul de automatizare. 16
2.4.2. Schema bloc 17
2.4.3. Schema desfăşurată electrică 18
2.4.4. Algoritmul automatului 18
2.4.5. Simularea funcționării automatului. 19
2.4.6. Structura sistemului automat. 22
2.4.7. Conducerea de la distanță. 22
2.4.8. Intrările şi ieşirile automatului. 23
2.4.9. Tehnologia de realizare a automatului. 23
2.4.10. Referința şi eroarea maximă. 23
2.4.11. Perioada de oscilației 24
2.4.12. Semnalizarea. 24
2.4.13. Rolul operatorului. 24
2.5. SISTEM CU AUTOMAT PROGRAMABIL LOGIC - APL. 24
2.5.1. Schema tehnologică cu echipamentul de automatizare. 24

2.5.2. Schema desfăşurată electrică. *25*
2.5.3. Programul. *26*
2.6. SISTEM CU REGULATOR BIPOZIȚIONAL - 2P. 26
2.6.1. Schema tehnologică cu echipamentul de automatizare. *27*
2.6.2. Schema bloc *28*
2.6.3. Eroarea maximă. *28*
2.6.4. Legătura inversă negativă. *29*
2.6.5. Algoritmul bipoziţional - 2P. *29*
2.7. SISTEM CU REGULATOR PROPORȚIONAL - P 29
2.7.1. Algoritmul proporţional - P. *30*
2.7.2. Constanta de proporționalitate a compensatorului *30*
2.7.3. Schema tehnologică cu echipamentul de automatizare. *31*
2.7.4. Schema bloc *31*
2.7.5. Intrarea treaptă şi răspunsul indicial. *32*
2.7.6. Regimul tranzitoriu şi regimul staționar. *32*
2.7.7. Răspunsul indicial în regim staționar. *33*
2.7.8. Eroarea indicială în regim staționar. *34*
2.7.9. Exemplu numeric. *34*
2.7.10. Stabilitatea şi robustețea. *35*
2.7.11. Banda de proporționalitate a compensatorului. *35*
2.8. SISTEM CU REGULATOR DIRECT. 37
2.9. SISTEM CU REGULATOR PROPORȚIONAL INTEGRAL DERIVATIV – PID 38
2.9.1. Algoritmul PI. *38*
2.9.2. Algoritmul PD. *40*
2.9.3. Algoritmul PID. *40*
2.10. SISTEM CU CALCULATOR. 41
2.10.1. Conducerea de supraveghere şi achiziția datelor *42*
2.10.2. Reglarea PID numerică *44*
2.10.3. Autotestarea şi autoacordarea. *46*
2.10.4. Acordarea automată adaptivă. *46*
2.10.5. Configurarea asistată de calculator. *46*

2.10.6. Sisteme de securitate la efracţie şi siguranţă la foc *46*
2.10.7. Sisteme domotice. *46*
2.11. MODELE CU VARIABILE DE STARE ALE SISTEMELOR 47
2.12. MODELE ÎN DOMENIUL FRECVENȚĂ ALE SISTEMELOR 56
2.12.1. Scurt istoric *57*
2.12.2. Semnale sinusoidale. Funcția de transfer. *58*
2.12.3. Propagarea semnalelor sinusoidale prin sisteme liniare *59*
2.12.4. Funcția de transfer a filtrului analogic de ordin unu. *60*
2.12.5. Funcția de transfer a filtrului analogic de ordin doi *64*
2.12.6. Semnale periodice. Seria Fourier *67*
2.12.7. Semnale impuls. Transformarea Laplace. Funcția de transfer. *67*
2.12.8. Semnificații fizice în domeniul frecvență *74*
2.12.9. Răspunsul sistemelor la semnale impuls în domeniul frecvență *74*
2.13. NOȚIUNI DE PROIECTRE A SISTEMELOR AUTOMATE 77
2.13.1. Proiectarea funcțională *77*
2.13.2. Proiectarea constructivă *80*

3. SISTEME LOGICE COMBINAȚIONALE - SLC 83

3.1. CODURI 85
3.2. ANALIZA SISTEMELOR LOGICE COMBINAȚIONALE 87
3.2.1. Sistem logic combinațional tip ŞI *88*
3.2.2. Sistem logic combinațional tip SAU *89*
3.2.3. Sistem logic combinațional tip NU *90*
3.2.4. Relații logice caracteristice sistemelor logice combinaționale *91*
3.3. SINTEZA SISTEMELOR LOGICE COMBINAȚIONALE 92
3.3.1. Metoda formei disjunctive canonice *92*
3.3.2. Metoda diagramei Karnaugh *95*
3.4. TESTAREA SISTEMELOR LOGICE COMBINAȚIONALE 100
3.5. IMPLEMENTAREA SLC CU AUTOMATE PROGRAMABILE LOGICE 102
3.6. IMPLEMENTAREA SLC CU CIRCUITE ELECTRONICE INTEGRATE 104
3.7. ASPECTE CONSTRUCTIVE ALE SISTEMELOR LOGICE COMBINAȚIONALE 105
3.7.1. Construcția releului electromagnetic *106*

3.7.2. Elemente reale ale sistemelor logice combinaţionale 108
3.7.3. Hazardul combinaţional. 109

4. SISTEME AUTOMATE CU EVENIMENTE DISCRETE 110

4.1. MODELAREA SISTEMELOR CU EVENIMENTE DISCRETE 111
4.2. DEFINIREA SISTEMELOR DISCRETE LOGICE. 116
4.3. TIPURI DE REŢELE PETRI. 117
4.3.1. Reţele Petri autonome. 117
4.3.2. Reţele Petri interpretate. 117
4.3.3. Reţele Petri temporizate. 117
4.4. ANALIZA STRUCTURALĂ A SISTEMELOR CU EVENIMENTE DISCRETE. 118
4.4.1. Structuri tip folosite la modelarea cu reţele Petri 118
4.4.2. Reţele Petri ordinare. 118
4.4.3. Reţele Petri pure 119
4.4.4. Poziţia sursă sau receptor. 119
4.4.5. Tranziţia validată. 119
4.4.6. Tranziţia declanşată. 120
4.4.7. Conflictul structural şi conflictul efectiv al tranziţiilor. 120
4.4.8. Interblocarea prin interpretare. 121
4.5. ANALIZA COMPORTAMENTALĂ A SISTEMELOR CU EVENIMENTE DISCRETE. 121
4.5.1. Ecuaţia de stare. 121
4.5.2. Graful marcajelor accesibile. 122
4.6. PERFORMANŢELE SISTEMELOR CU EVENIMENTE DISCRETE. 122
4.6.1. Reversibilitatea 123
4.6.2. Mărginirea şi siguranţa. 124
4.6.3. Viabilitatea. 126
4.7. SISTEME CU EVENIMENTE DISCRETE ŞI EVOLUŢIE PARALELĂ 127
4.8. MAŞINA DE STARE. 130
4.9. GRAFCETUL. 132
4.9.1. Trecerea de la reţeaua Petri la grafcet. 132
4.9.2. Etape, tranziţii şi legături orientate. 133
4.9.3. Interpretarea tranziţiilor. 134

4.9.4. Interpretarea etapelor. *134*
4.9.5. Reguli de evoluție în grafcet. *135*
4.9.6. Structuri folosite la modelarea cu grafcet a sistemelor discrete logice... *135*
4.9.7. Compararea grafcetului cu rețeaua Petri. *136*
4.10. SINTEZA SISTEMELOR DISCRETE LOGICE. 138
4.11. IMPLEMENTAREA SISTEMELOR DISCRETE LOGICE. 139
4.12. AUTOMATE ELEMENTARE CU CONTACTE ŞI RELEU. 139
4.12.1. Analiza structurală. *140*
4.12.2. Analiza comportamentală. *141*
4.12.3. Automatul elementar cu basculare. *142*
4.12.4. Automatul elementar cu prioritate la oprire. *146*
4.12.5. Automatul elementar cu prioritate la pornire. *149*
4.12.6. Automatul elementar cu neschimbarea stării. *151*
4.12.7. Automatul elementar pentru reglarea bipoziţională. *153*
4.13. AUTOMATE ELEMENTARE CU CIRCUITE INTEGRATE 156
4.13.1. Bistabilul RS. *156*
4.13.2. Bistabilele SR şi SRC. Sincronizarea. *158*
4.13.3. Bistabilul SCR Master – Slave. *159*
4.13.4. Bistabilul JKC. *160*
4.13.5. Bistabilul D. *160*
4.13.6. Bistabilul T. *161*
4.14. PROIECTAREA AUTOMATELOR IMPLEMENTATE CU CONTACTE ŞI RELEE 162
4.14.1. Schema tehnologică şi schema bloc *162*
4.14.2. Caietul de sarcini al automatului *165*
4.14.3. Analiza structurală şi comportamentală *165*
4.14.4. Sinteza automatului cu contacte şi relee. *166*
4.14.5. Implementarea automatului cu contacte şi relee. *169*
4.15. PROIECTAREA AUTOMATELOR IMPLEMENTATE CU BISTABILE 171
4.15.1. Sinteza şi implementarea automatului cu bistabile tip D *171*
4.15.2. Sinteza şi implementarea automatului cu bistabile JKC. *174*
4.15.3. Problema iniţializării sistemului discret logic. *175*

4.16. PROIECTAREA AUTOMATELOR IMPLEMENTATE CU APL 179
4.16.1. Metoda Grafcet 184
4.16.2. Metoda listei de instrucţiuni 184
4.16.3. Metoda schemei desfăşurate 191
4.16.4. Schema electrica desfăşurată pentru APL 193
4.16.5. Metoda de proiectare pentru APL aplicată la contacte şi relee 194
4.16.6. Proiectarea automatelor pentru reţele Petri simple. 196
4.16.7. Proiectarea automatelor folosind automate elementare tip RS 196
4.17. ASPECTE CONSTRUCTIVE ALE AUTOMATELOR CU EVENIMENTE DISCRETE 197
4.17.1. Schema tehnologică cu echipamentul de automatizare 197
4.17.2. Schema electrică desfăşurată 201
4.17.3. Tipuri constructive de automate programabile logice 206
4.18. AUTOMATE PENTRU MOTOARE 208
4.18.1. Alimentarea automatelor pentru motoare 208
4.18.2. Automat pentru pornirea directă 209
4.18.3. Automat pentru pornirea stea triunghi 210
4.19. SUBSISTEME SPECIFICE ALE AUTOMATELOR 211
4.19.1. Sisteme de alimentare 211
4.19.2. Sisteme de achiziţie a intrărilor din proces 212
4.19.3. Sisteme de conducere automată 212
4.19.4. Sisteme de conducere manuală 213
4.19.5. Sisteme de semnalizare automată 213
4.19.6. Sisteme de execuţie şi de protecţie automată 214
4.20. APLICAŢII 214
4.20.1. Automate simple cu APL 214
4.20.2. Automat cu APL Tl5 pentru banda rulantă 222
4.20.3. Automat cu APL pentru motorul reversibil cu iniţializare 227
4.20.4. Automat cu contacte şi relee pentru ascensor 233
4.20.5. Automat cu APL pentru anclanşarea automată a rezervei tehnologice. 241

5. SISTEME DE CONDUCERE ÎN CAZ DE PERICOL 246

5.1. CONCEPTELE SISTEMELOR DE SECURITATE LA EFRACŢIE ŞI SIGURANŢĂ LA FOC 248

5.2. CENTRALE DE CONDUCERE ÎN CAZ DE PERICOL 251
5.3. TRADUCTOARE PENTRU SISTEME DE SECURITATE LA EFRACȚIE 252
5.3.1. Detectoare pasive în infraroşu - PIR 252
5.3.2. Monitoare video de mişcare 252
5.3.3. Controlere de acces 253
5.3.4. Sisteme de identificare cu frecvență radio 256
5.4. TRADUCTOARE PENTRU SISTEME DE SIGURANȚĂ LA FOC 257
5.4.1. Butoane manuale de semnalizare. 259
5.4.2. Detectoare de temperatura 260
5.4.3. Detectoare de fum cu ionizare 262
5.4.4. Detectoare de fum optice 264
5.4.5. Detectoare de fum fotoelectrice cu obturarea luminii 264
5.4.6. Detectorare de fum fotoelectrice cu difuzarea luminii 265
5.4.7. Amplasarea detectoarelor de pericol la incendiu 266
5.5. SISTEM SIMPLU DE SECURITATE 270
5.6. SISTEMUL MICROSAM 271
6. BIBLIOGRAFIE 275

1. Prefață

Acest curs este destinat studenților de la Facultatea de Instalații şi de la Departamentul de Automatică şi Informatică Aplicată şi o denumire mai riguroasă a sa ar fi *Sisteme de conducere automată a echipamentelor, instalațiilor şi serviciilor din locuințe şi clădiri*[1]. Se foloseste frecvent si denumirile de *Sisteme automate* sau *Automatizarea instalațiilor*. Denumirea *Sisteme automate* reprezintă o prescurtare avantajoasă si reflectă totodată faptul că sistemele automate din clădiri sunt aceleaşi cu cele din industrie sau alte domenii, diferențele fiind de nuanțe şi accente[2]. De exemplu, în industrie multe utilaje şi echipamente funcționează într-un mediu prietenos, programat şi cu tehnologii alese astfel încât perturbațiile să fie cât mai mici. Cerințele referitoare la performanțe şi fiabilitate sunt foarte ridicate. Funcționarea instalațiilor din clădiri se realizează într-un context mai puțin prietenos deoartece depinde foarte mult de activitatea zilnică a omului şi perturbațiile mediului înconjurător. Pe de altă parte cerințele referitoare la fiabilitate şi performanțe nu sunt tot atât de riguroase ca în industrie ceea ce conduce la costuri mai mici. Din aceste motive în automatizarea instalațiilor se implementează cel mai repede ultimele noutăți din teoria şi tehnologia conducerii automate. Pe de altă parte sistemele de conducere automată din transporturi şi armată funcționează totodată, spre deosebire de cele din industrie sau clădiri, atât în medii foarte neprietenoase cât şi cu condiții stricte referitoare la performanțe şi fiabilitate.

Acest curs prezentat în formă electronică, format pdf, este actualizat frecvent. Din acveastă cauză este importantă versiunea şi anul apariției.

---

[1] La Facultatea de Instalații este încetățenită denumirea de *Automatizarea instalațiilor*. Acest nume pune accentul pe *acțiunea* de conducere automată şi lasă în planul doi analiza şi proiectarea componentelor sistemelor automate. In engleză se foloseşte: *Home and Building Automation*

[2] De exemplu, sistemele de conducere automată din clădiri funcționează, spre deosebire de cele din industrie, mai mult în regim de reglare automată şi mai puțin în regim de urmărire sau filtrare.

2. Introducere în tehnica sistemelor automate

Noțiunea de ***sistem*** se referă întotdeauna la ***o mulțime de obiecte, acțiuni sau idei aflate în interacțiune pentru un anumit scop***. In această lucrare vom folosi această denumire numai în legătură cu o mulțime de obiecte fizice. Dacă ne referim la obiectele tehnice dintr-o clădire, care împreună asigură anumite servicii, sistemul mai este cunoscut şi sub numele de ***instalație***[3]. Sistemele cu conducere automată, pe scurt ***sistemele automate***[4], sunt un tip particular de sisteme a căror scop este să funcționeze fără intervenția omului realizând anumite performanțe în diferite regimuri de funcționare: urmărirea referinței dorite, reglarea pentru înlăturarea perturbațiilor, filtrarea zgomotelor. Sistemele automate au ***elemente*** şi ***structuri*** specifice. Un ***sistem automat*** tipic este format din următoarele elemente componente: 1) elemente de conducere (***automate, termostate, presostate, regulatore, controlere, calculatoare, etc***), 2) elemente de execuție (***servomotoare, pompe, robinete, rezistoare, etc)***, 3) elemente de măsurare (***traductoare, detectoare***), 4) elemente conduse (**echipamente,** ***instalatii, procese)***. Structura este formată din mulțimea relațiilor dintre elemente. Structura tipică a sistemelor automate o reprezintă ***bucla*** care conține o legătură inversă negativă (***feedback, reacție)*** de la ieşirea la intrarea sistemului. În esență un sistem este automat dacă conține 1) ***feedback negativ*** şi 2) ***memorare***[5]. Elementele componente ale sistemului automat pot fi de natură termică, hidraulică, electrică, mecanică, chimică sau pneumatică. Principalele probleme referitoare la sistemele automate sunt: analiza, sinteza, testarea, optimizarea, identificarea şi proiectarea. Scopul sistemelor automate prezentate în continuare îl constitue conducerea, semnalizarea, protecția, alarmarea şi optimizarea performanțelor referitoare la confort, consum de energie şi material, pericole şi sensibilitatea la defecțiuni. Conceptele şi metodele sistemelor automate vor fi prezentate în continuare pe scurt plecând de la sistemele cu conducere manuala.

---

[3] Noțiunea de *sistem* este utilizată într-un cadru mai larg. Se poate folosi, de exemplu, termenul de *sistem de încălzire* sau echivalentul său *instalație de încălzire*. Termenul *sistem de transport* nu are însă echivalent.

[4] Instalațiile automate.

[5] Memorarea poate fi realizată prin mai multe metode, una dintre ele fiind metoda feedbackului pozitiv.

2.1. Istoria sistemelor automate

Războiul de țesut cu cartele perforate, primul sistem de conducere automată cu evenimente discrete, a fost inventat de Joseph-Marie Charles Jacquard în 1800.

Primul sistem de reglare automată, în sens modern, a folosit regulatorul mecanic centrifugal al lui James Watt (1769) prezentat în Fig. 2.1. Acest aparat regla automat viteza unei maşini cu aburi cu ajutorul unui robinet de reglare a debitului de abur. Creşterea vitezei maşinii cu aburi provoca ridicarea bilelor datorită forței centrifuge. Are loc o reacție negativă care închide robinetul aducând viteza la valoarea dorită de referință. Acest tip de regulator este folosit şi astăzi. Analiza făcută de Clark Maxwell acestui sistem în 1868 a condus la primul criteriu de stabilitate a sistemelor care solicita prezența polilor sistemului cu bucla închisă în semiplanul stâng.

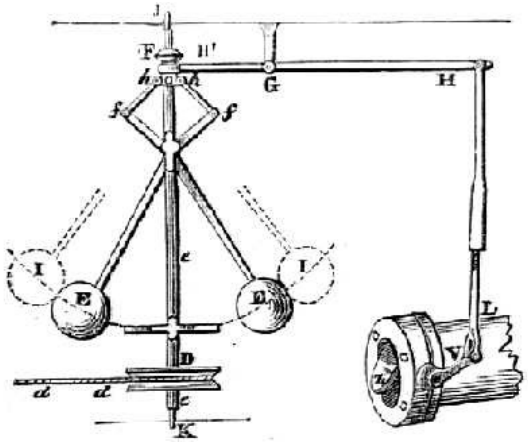


Fig. 2.1 Regulatorul centrifugal al lui Watt.

Termostatul cu bimetal a fost inventat[6] în 1885. Harold Stephen Black[7] a inventat în 1923 primul amplificator electronic cu reacție negativă iar Hendrik Wade Bode a pus bazele teoriei sistemelor cu reacție. În anii celui de al doilea război mondial s-a inventat pilotul automat[8], servo - sistemele de urmărire a poziției pentru nave şi tunuri, regulatorul PID şi s-a introdus metoda de analiză bazată pe modelele funcționale[9] şi criteriul de stabilitate Nyquist.

---

[6] Inventatorul a fost Al Butz care a pus bazele companiei Honeywell

[7] De la American Telephone and Telegraph. Acordarea brevetului de invenție a durat nouă ani deoarece specialiştii nu credeau că sistemul funcționază.

[8] Primul pilot automat inventat de Honeywell a fost produs în peste 110000 exemplare.

[9] Modele caracterizate de funcția de transfer.

Metoda locului rădăcinilor a lui Evans a apărut în anii de după 1950 odată cu conducerea optimală şi conducerea adaptivă. Conducerea numerică directă[10] şi modele cu variabile de stare au fost folosite începând cu anii 1960. Între anii 1970 şi 1980 au apărut sistemele de conducere cu microcalculatoare, cu automate programabile logice, sistemele de conducere distribuite[11] şi tehnologia de autoacordare şi acordare a regulatoarelor.

2.2. Sistem cu conducere manuală

Cele mai simple sisteme cu conducere automată reproduc modul de conducere manuală. Vom prezenta un exemplu simplu şi pentru descrierea acestuia vom folosi principalele concepte folosite la sistemele automate: *schema tehnologică cu echipamentul de automatizare, schema bloc şi algoritmul de conducere.*

2.2.1. Schema tehnologică.

Să considerăm un sistem de încălzire a apei menajere dintr-o clădire. Reprezentarea obiectelor componente se face cu ajutorul unui desen *simplificat* numit ***schemă tehnologică***, ca în *Fig. 2.2*. În acest caz instalaţia este compusă din mai multe obiecte tehnice: pompa P, rezervorul, rezistorul de încălzire R, termometrul Z, rigla M1 de măsură a nivelului, contactul S1 de alimentare al rezistorului şi robinetele V1 şi V2. Ele formează un sistem deoarece *interacţionează* într-un anumit *scop*: obţinerea apei calde.

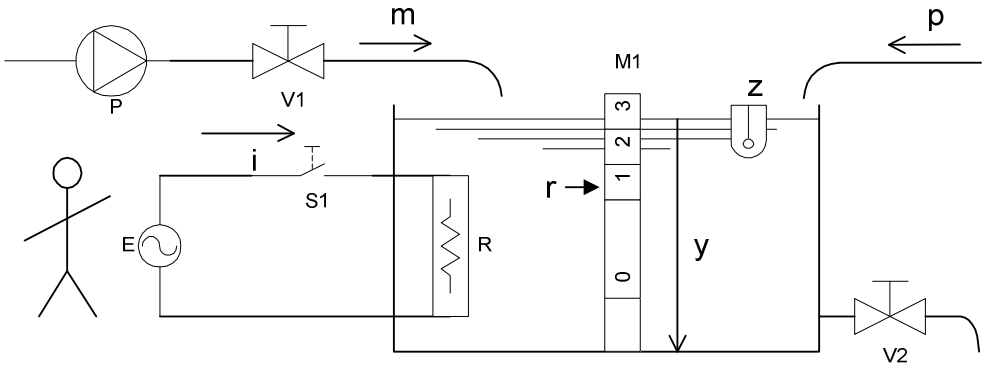


Fig. 2.2 Schema tehnologică a sistemului (instalaţiei) de încălzire a apei menajere.

In instalaţie există procese de natură fizică diferită: hidraulică, termică, electrică şi mecanică. Din această cauză la proiectarea instalaţiei se definesc sisteme separate, mai simple decât cel din *Fig. 2.2*. De exemplu, sistemul termic poate cuprinde numai trei obiecte: rezistorul de încălzire, termometrul şi rezervorul.

[10] DDC - Direct digital control.

[11] Primul sistem a fost elaborat de firma Honeywell

2.2.2. Schema bloc

Instalația din *Fig. 2.2* este condusă manual. Operatorul observă temperatura z şi nivelul y al apei şi acționează butonul de încălzire S1 şi robinetul V1 pentru a obține valorile dorite. Din punctul de vedere al conducerii instalației toate amănuntele din *Fig. 2.2* nu sunt necesare. Se renunță la ele şi se defineşte un ***sistem de conducere manuală*** sub forma unei ***scheme bloc*** ca în *Fig. 2.3*. Sistemul are în cazul acesta o singură componentă denumită bloc. Acesta este o cutie neagră[12] care ascunde amănuntele din *Fig. 2.2* şi pune în evidență numai intrările şi ieşirile sistemului. Dacă se poate considera că fiecare intrare influențează numai o singură ieşire[13] atunci se poate defini o schemă bloc pentru sistemul de conducere a nivelului prezentată în *Fig. 2.4* cu o singură intrare m şi o singură ieşire y. Se observă în aceasta că blocul se referă numai la procesul hidraulic de umplere al instalației şi nu cuprinde toate componentele sistemului din *Fig. 2.2* care include şi încălzirea apei. Schema tehnologică corespunzătoare este prezentată în *Fig. 2.5*. *Sistemele cu o intrare şi o ieşire*[14] vor fi singurele care vor fi luate în considerare în continuate

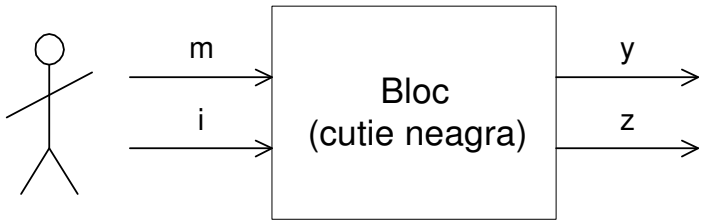


Fig. 2.3 Schema bloc a sistemului de conducere manuală pentru instalația de încălzire.

2.2.3. Algoritmul de conducere manuală.

Conducerea manuală a procesului de umplere din instalația din Fig. 2.5 se face in modul următor. Operatorul observă pe rigla M1 valoarea actuală y şi valorile dorite (referințe) m (valoarea minimă) şi M (valoarea maximă) a nivelului în rezervor. Pentru situația din *Fig. 2.5* se obține y=3, m=1, şi M=3. Operatorul face câteva comparații logice şi în funcție de rezultatele obținute execută câteva acțiuni. În cuvinte obişnuite comportarea operatorului este următoarea: dacă nivelul este mai mare decât nivelul maxim M închide robinetul V1 iar dacă nivelul este mai mic decât nivelul minim m deschide robinetul V1. Robinetul V1 este normal

[12] Black box.

[13] In realitate fiecare intrare influențează ambele ieşiri. De exemplu debitul m influențează într-o oarecare măsură şi temperatura z, nu numai nivelul y.

[14] SISO – Single Input Single Output

închis. Deschiderea lui se face cu ajutorul electromagnetului S1 prin acționarea contactului u[15].

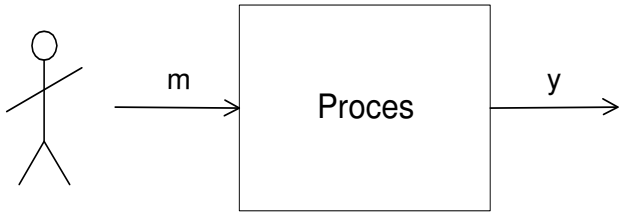


Fig. 2.4 Schema bloc a sistemului de conducere manuală a nivelului din instalația de încălzire.

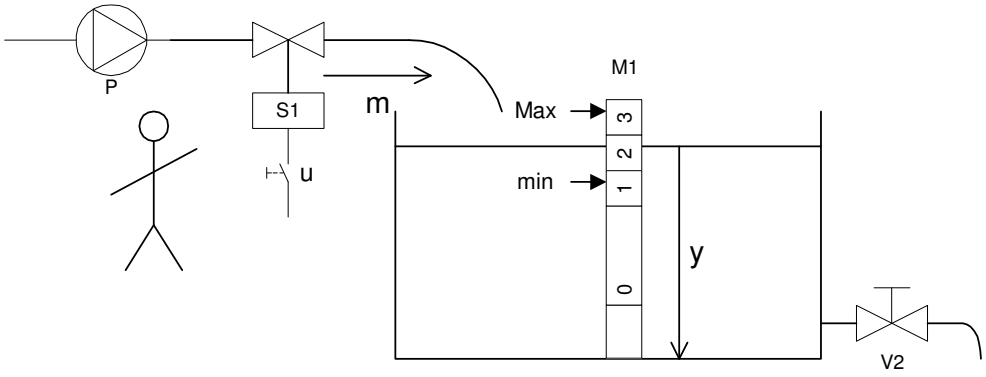


Fig. 2.5 Schema tehnologică a sistemului pentru reglarea manuală a nivelului.

2.3. Semnale

Legătura dintre elementele componente ale sistemelor de conducere automate se face prin intermediul semnalelor.

2.3.1. Semnale discrete logice

Semnalele discrete logice au numai două valori: adevărat (valoarea logică 1) şi fals (valoarea logică 0) şi pot fi realizate fizic în diferite variante, de exemplu cu ajutorul contactelor şi releelor care au două stări: acționat şi neacționat

[15] Regulator manual.

2.3.2. Semnale continue

Semnalele continue au o variație continuă în timp de tip periodic, impuls sau aleatoriu. Semnalul treaptă unitară, semnalul rampă unitară sau semnalul sinusoidal sunt semnale de tip continuu

2.3.3. Semnale eşantionate

Sistemele de conducere cu calculatoare folosesc semnale eşantionate în timp cu o perioada constantă.

2.4. Sistem cu automat cu contacte şi relee.

Operațiile efectuate de operatorul din Fig. 2.5 sunt în esență următoarele:

1) Măsurarea nivelului,
2) Realizarea algoritmului de conducere,
3) Execuția comenzii.

Lista 2.1 Algoritmul pentru reglarea manuală.

Se aplică una dintre regulile:

Regula 1: Dacă (y>Max)=**Adevărat**, atunci (k=acționat)=**Fals,** adică se lasă închis robinetul V1.

Regula 2: Daca (y>min)=**Fals**, atunci (k=acționat)=**Adevărat,** adică se deschide robinetul V1.

Regula 3: Daca (y>min) **ŞI NU**(y>Max)=**Adevărat**, atunci se repetă acțiunea precedentă. Pentru a putea face aceasta se **memorează** de fiecare dată valoarea variabilei logice (k=acționat) şi deci se ştie care a fost ultima acțiune de deschidere sau închidere a robinetului V1.

2.4.1. Schema tehnologică cu echipamentul de automatizare.

Instalația în care nivelul rezervorului este reglat automat este prezentată în Fig. 2.6. Măsurarea nivelului este realizată de un *traductor* format dintr-un plutitor, un cablu şi o contragreutate. Contactul F7 cu memorie mecanică care este legat de un palpator Q formează *automatul*. Contactul se deschide când nivelul depăşeşte valoarea maximă a şi se închide când nivelul este mai mic decât valoarea minimă b. Atunci când nivelul se află între aceste două poziții automatul memorează ultima valoare (acționat, neacționat) a contactului F7. Automatul F7 comandă *elemetul de execuție* format din motorul electric M şi pompa. Siguranțele F1 protejează automat linia de alimentare L1, L2 şi L3, iar comutatorul Q1 protejează automat motorul M.

Instalațiile automatizate sunt reprezentate simplificat cu ajutorul schemei tehnologice cu echipamentul de automatizare. De exemplu, instalației din Fig. 2.6 îi corespunde schema din Fig. 2.7.

Aparatele sunt prezentate în schema tehnologică cu echipamentul de automatizare din Fig. 2.7 sub forma unor cercuri numerotate de la 1 la 3. Tipul fiecărui aparat este specificat deasupra numărului aparatului sub forma unui grup de două litere[16]. Prima literă este L şi indică faptul că toate aparatele sunt destinate conducerii nivelului (în engleză *level*)[17]. A doua şi a treia literă indică funcțiunea aparatului. Aparatul 1 are litera T care indică funcția de transmisie la distanță, aparatul 2 are litera K, adică este o stație de comandă cu contacte şi relee şi litera A pentru funcțiunea de semnalizare, iar aparatul 3 este un element de execuție deoarece are litera Z în poziția a doua[18]. Cercurile cu o linie orizontală indică aparate montate în camera de comandă, de exemplu cel cu numărul 3, iar cercurile fără linie orizontală desemnează aparate montate pe instalație, cum este traductorul de nivel cu numărul 1, Fig. 2.7. Distanța dintre locurile în care sunt montate aparatele 1 şi 2 poate fi mare, nedepăşind însă 1000 metri.

---

[16] STAS 6755-81.

[17] Alte litere frecvent întrebuințate sunt următoarele: B-flacără, E-tensiune electrică, F-debit, H-acționare manuală, K-timp sau program, L-nivel, M-umiditate, O-semnificație la alegere, P-presiune, Q-cantitate, S-viteză sau frecvență, T-temperatură, W-greutate sau forță, Z-poziție.

[18] Alte semnificații ale literei din poziția a doua care indică funcțiunea aparatului sunt: A-alarmă, C-reglare, E-element primar al traductorului, I-indicare, L-lampă pilot, N-semnificație la alegere, R-înregistrare, Z-element de execuție.

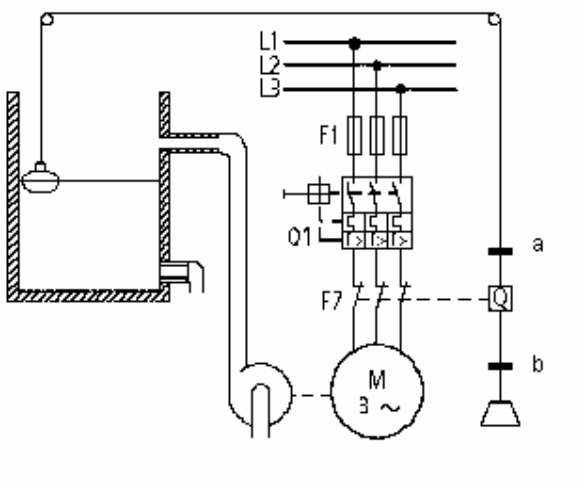


Fig. 2.6 Schema tehnologică pentru reglarea automată anivelului folosind drept element de executie o pompă.

2.4.2. Schema bloc

O altă formă de prezentare a sistemului automat din Fig. 2.7 este schema bloc din *Fig. 2.9*. Dacă blocurile din schema tehnologică prezentate sub formă de cercuri reprezintă aparate distincte constructiv, dreptunghiurile din schema bloc colectează toate aparatele cu aceiaşi funcţiune. De exemplu, amplificatorul 3 şi electromagnetul S1 din schema tehnologică sunt prezentate sub forma unui singur bloc în *Fig. 2.9*: elementul de execuție. Operatorul din schema bloc introduce referința sistemului automat, adică valorile dorite pentru nivelul maxim şi nivelul minim.

2.4.3. Schema desfăşurată electrică

Inginerul automatist trebuie să proiecteze automatul 2 din Fig. 2.7. O piesă importantă a proiectului este schema desfăşurată electrică cu contacte şi relee. O variantă simplificată este prezentată în Fig. 2.8. Schema are forma unei scări[19] culcate în care "treptele" reprezintă circuite numerotate şi conectate între barele de alimentare cu tensiune electrică. De exemplu în circuitul 10 se găseşte bobina releului K şi contactul normal închis a. Diagrama de contacte a releului desenată sub numărul circuitului arată că acesta are două contacte normal deschise, unul situat în circuitul 20 iar altul în circuitul 30[20]. In circuitul 30 al schemei apare contactorul K1. Acesta este un releu de forță, cu rol de amplificare a curentului furnizat de automat astfel încât acesta să aibă o valoare suficient de mare pentru acționarea electromagnetului S1 din circuitul 40. Semnalizarea umplerii rezervorului este făcută de lampa H1 din circuitul 50.

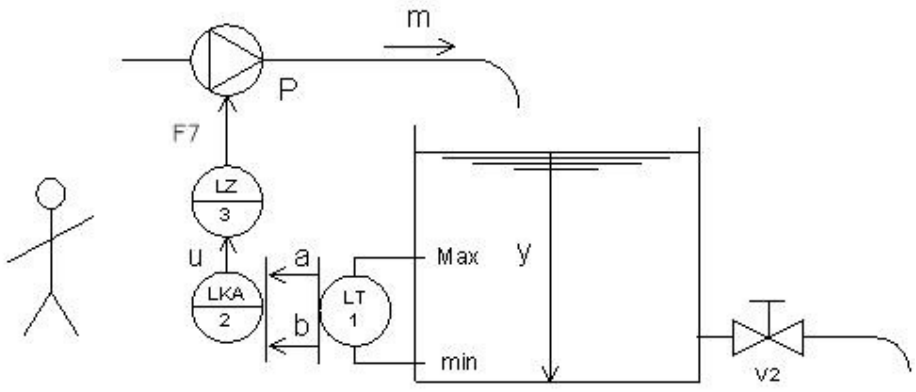


Fig. 2.7 Schema tehnologică cu echipamentul de automatizarea a sistemului cu automat cu contacte şi relee pentru reglarea nivelului si element de executie pompa.

2.4.4. Algoritmul automatului

În orice prezentare tehnică a unui sistem automat trebuie să se indice algoritmul de conducere. Există foarte moduri în care poate fi condusă o instalație. Din motive practice se preferă de cele mai multe ori câteva variante de algoritm clasice. În cazul de față, algoritmul prezentat în Lista 2.1 pentru conducerea manuală şi aplicat şi la conducerea automată este de

[19] Ladder network, în engleză.

[20] Detalii privind întocmirea schemelor desfăşurate electrice în IEC 1082-1 (decembrie 1982), STAS 7070-74 şi manualele de proiectare a instalațiilor de automatizare.

tip *automat secvențial bipozițional*[21], sau prescurtat **tip 2P**. Potrivit acestui algoritm instalația automată se găseşte numai în două poziții[22]: umplere sau golire. Conducătorul manual sau automat basculează sistemul între aceste poziții astfel încât nivelul să se mențină între limitele **min** şi **Max.**

2.4.5. Simularea funcționării automatului.

Descrierea funcționării sistemului automat poate fi făcută în multe feluri, dintre care menționăm: cu ajutorul formulelor, a programelor pe calculator sau prin cuvinte cu ajutorul unei simulări intuitive în imaginație[23]. Ultima variantă este preferată tehnicienilor de întreținere a automatelor cu contacte şi relee. Să exemplificăm această metodă de simulare pentru sistemul automat descris de Fig. 2.7 şi Fig. 2.8.

Simularea porneşte cu sistemul având condiții inițiale zero, adică rezervorul este gol şi toată instalația nu este alimentată cu energie electrică.

Se alimentează instalația cu energie electrică. Pompa P porneşte şi traductorul, automatul şi elementul de execuție intră în funcțiune.

Traductorul măsoară nivelul în rezervor. El are doi electrozi montați la nivelul min şi nivelul Max, Fig. 2.7. În momentul în care apa acoperă electrodul min apare un curent între acesta şi fundul rezervorului[24] care acționează contactul a. Asemănător, atunci când apa acoperă electrodul Max se acționează contactul b.

În situația existentă la începutul simulării pompa abia a pornit, nivelul y este mai mic decât m şi nici un contact a sau b nu este acționat.

Automatul primeşte semnalele a şi b sub formă de contacte şi în funcție de valoarea lor trimite o comandă elementului de execuție. În situația inițială nici un contact nu este acționat şi în schema din Fig. 2.8 se petrec mai multe acțiuni. De la borna +24V a tensiunii de alimentare circulă un curent prin circuitul 10 deoarece contactul a, normal închis, nu este acționat[25]. Releul K îşi acționează toate contactele sale situate, conform diagramei de contacte, în circuitele, 20, 30 şi 50.

Contactorul K1 acționează şi lampa H1 se aprinde semnalizând umplerea rezervorului. Contactul lui K1 din circuitul 40 se acționează, deci fiind normal deschis se închide şi

---

[21] Mai exact bipozițional cu histerezis de lățime 2h = Max - min.

[22] Stări.

[23] Din păcate această metodă nu mai este utilă în cazul sistemelor automate mai complicate descrise în continuare. În aceste cazuri sistemul este descris cu ajutorul funcțiilor de transfer sau a programelor de simulare pe calculator.

[24] Rezervorul este metalic, altfel se montează un electrod suplimentar la fundul său.

[25] Traseul curentului este următorul: borna +24V, contactul a, bobina releului K, borna –24V.

electromagnetul S1 intră în funcţiune deschizând robinetul V1. Rezervorul începe să se umple şi nivelul creşte. La un moment dat apa acoperă electrodul min al traductorului, deci y>min, şi contactul a este acţionat. Fiind normal închis contactul a se deschide.

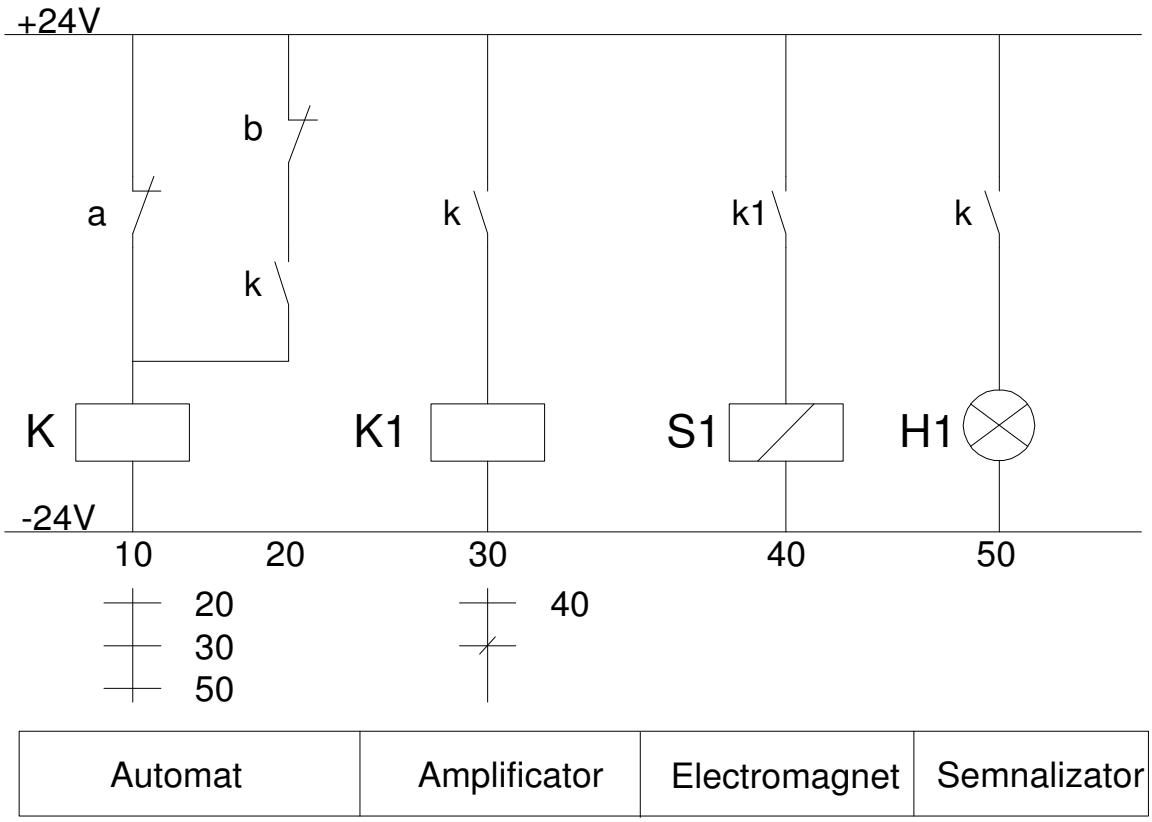


Fig. 2.8 Schema desfăşurată electrică simplificată pentru automatul 2, elementul de execuţie 3 şi electromagnetul S1 din Fig. 2.7.

Apare o nouă situaţie pentru automat pe care acest trebuie să o interpreteze. Circuitul 10 prin care a fost atras releul K se întrerupe datorită deschiderii contactului a. Cu toate acestea releul K rămâne acţionat prin circuitul 20 datorită faptului că contactul k este închis. Ne reamintim că acest contact a fost închis la pornirea instalaţiei când releul K s-a atras prin circuitul 10. Acum, când acest circuit s-a întrerupt contactul k din circuitul 20 menţine releul atras. Tehnicienii numesc acest contact de automenţinere. Observăm că releul îşi menţine singur starea precedentă cu ajutorul contactului său de automenţinere. Deoarece K rămâne atras nu se schimbă nimic din funcţionarea instalaţiei şi nivelul creşte în continuare.

La depăşirea nivelului maxim Max se acţionează contactul b care fiind normal închis în circuitul 20 se deschide. În această situaţie releul K este nu mai este acţionat deoarece

ambele circuite 10 ş20 sunt întrerupte. În consecință K1 şi S1 nu mai sunt acționate şi robinetul V1 este închis iar lampa H1 nu mai luminează. Nivelul începe să scadă.

La un moment dat nivelul scade sub valoarea maximă Max şi contactul b nu mai este acționat. În circuitul 20 contactul b revine în poziția de normal închis, dar releul K nu este acționat deoarece contactul său k din acelaşi circuit este deschis memorând situația sa anterioară. Robinetul V1 rămâne în continuare închis şi nivelul continuă să scadă.

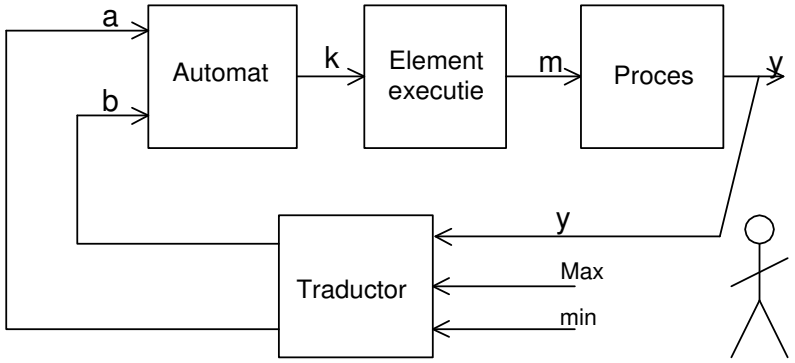


Fig. 2.9 Schema bloc a sistemului cu automat cu contacte şi relee din Fig. 2.7

Micşorarea nivelului sub valoarea min dată de poziția electrodului traductorului de nivel conduce la neacționarea contactului a, normal închis, şi reconstituirea circuitului 10 de alimentare a bobinei releului K. Ca urmare K1 şi S1 sunt acționate iar robinetul V1 este deschis şi lampa H1 luminează. Ciclul de funcționare se reia.

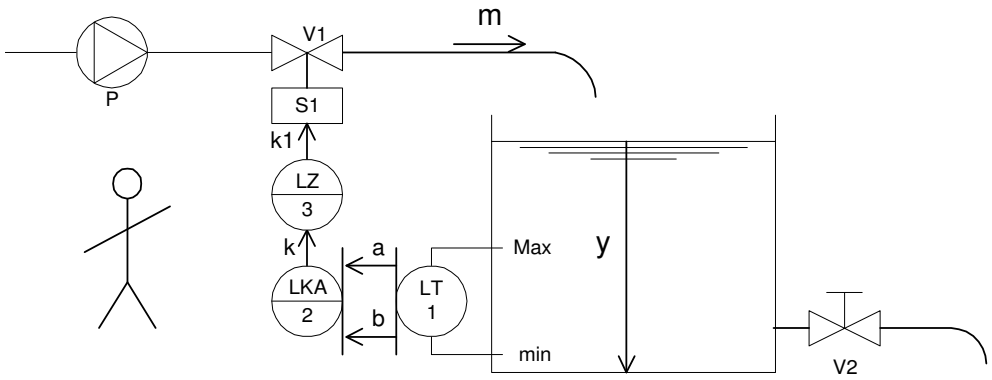


Fig. 2.10 Variantă a sistemului automat din Fig. 2.7.

Aceasta este simularea în imaginație a funcționării sistemului din Fig. 2.7 pentru reglarea automată a nivelului. Deşi este intuitivă şi simplu de efectuat această simulare nu ia în considerare toate situațiile posibile în funcționarea instalației. Să presupunem, de exemplu, că se opreşte alimentarea cu energie electrică a instalației datorită unei avarii. Nivelul apei în rezervor este cuprins între valoarea min şi Max a poziției electrozilor traductorului. În această

situaţie, la funcţionarea normală sistemul îşi menţine prin contactul de memorare k, circuitul 20 Fig. 2.8, starea anterioară de funcţionare. Cu alte cuvinte, dacă robinetul era deschis rămâne deschis iar dacă era închis rămâne închis. În această stare de avarie robinetul V1 este întotdeauna închis şi rezervorul se goleşte până când nivelul scade sub valoarea m a electrodului traductorului de nivel. Din acest moment se reia ciclul normal de funcţionare.

2.4.6. Structura sistemului automat.

Instalaţia din Fig. 2.7 este automatizată spre deosebire de instalaţia din Fig. 2.5 care este condusă manual. *Un sistem automat este caracterizat prin prezenţa* ***memoriei sau a legături inverse*** *(feedback, reacţie)*[26]. In felul acesta el evoluează în timp fără a mai fi necesară intervenţia omului. Pentru automatul cu contacte şi releu din Fig. 2.8 memorarea stării precedente a instalaţiei este realizată cu ajutorul contactului de automenţinere k care reprezintă tot o legătură inversă de la ieşirea releului la intrarea sa. Legătura inversă din Fig. 2.7 urmează următorul drum: ieşirea y (nivelul măsurat în metrii) a procesului reglat automat din instalaţie, aparatul 1 (traductorul), aparatul 2 (automatul), aparatul 3 (elementul de execuţie), electromagnetul S1, robinetul V1 şi intrarea u (debitul măsurat în m^3/h) a procesului. Ea poate fi observată mai bine în schema bloc a sistemului automat din *Fig. 2.9*. În limbaj ingineresc legătura inversă este caracterizată prin existenţa unei ***bucle***. Aparatele care fac parte dintr-o buclă se disting uşor în schema tehnologică deoarece au simbolul de identificare cu aceiaşi primă literă. De exemplu, în Fig. 2.7 există o buclă de reglare a nivelului iar simbolul celor trei aparate începe cu litera L.

2.4.7. Conducerea de la distanţă.

O altă caracteristică importantă a sistemelor automate, neobligatorie dar frecvent întâlnită, o constitue conducerea procesului de la o distanţă cuprinsă între câţiva metri şi 1000 metri. Şi instalaţia automată din Fig. 2.7 posedă această caracteristică deoarece aparatele 1 şi 3 sunt montate pe instalaţie iar aparatul 2, automatul cu contacte şi releu, este montat în camera de comandă[27] deoarece simbolul său are o bară orizontală.

---

[26] Termenul de feedback reprezintă denumirea engleză a legăturii inverse specifice sistemelor automate şi este folosit deseori ca atare şi în limba română. Termenul de reacţie pentru legătura inversă este inspirat din limba franceză şi este folosit mai ales în electronică.

[27] Faptul că este montat în camera de comandă rezultă din prezenţa unei linii între simbolul şi numărul său.

2.4.8. Intrările şi ieşirile automatului.

Semnalele cu care aparatele de automatizare comunică între ele reprezintă o caracteristică foarte importantă. Natura lor fizică şi tipul lor influențează rapiditatea sistemului automat, distanța de la care se poate face conducerea sistemului şi tipul aparatelor de automatizare folosite. Din Fig. 2.8 şi *Fig. 2.9* rezultă că intrările şi ieşirile automatului sunt semnale de tip contact. Distanța la care pot fi transmise aceste semnale este de ordinul zecilor de metrii.

2.4.9. Tehnologia de realizare a automatului.

Modul în care este implementat automatul influențează puternic multe caracteristici tehnice şi economice ale sistemului automat. Automatul din Fig. 2.8 este realizat cu contacte şi releu. Această tehnologie a fost şi încă mai este foarte răspândită la realizarea sistemelor automate mici şi ieftine. Exista o mare experiență în domeniul acestei tehnologii, acumulată pe parcursul zecilor de ani de folosință. Sistemele automate realizate cu ajutorul ei sunt robuste, ieftine, pot funcționa în medii ostile, sunt puțin sensibile la perturbații, pot fi realizate într-un număr mare de variante care să se adapteze la procese foarte diferite şi pot fi întreținute uşor de un personal cu o calificare nu prea înaltă. De exemplu, automatul din figura Fig. 2.8 se poate adapta foarte uşor la o altă schemă tehnologică, prezentată în Fig. 2.10, în care elementul de execuşie este un robinet acționat de un electromagnet.

2.4.10. Referința şi eroarea maximă.

Se poate defini o valoare medie a nivelului numită referință:

$$w = \frac{Max + \min}{2} \qquad (2.1)$$

şi eroarea maximă

$$e_{\max} = \frac{Max - \min}{2} \qquad (2.2)$$

sau eroarea maximă procentuală

$$\frac{e_{\max}}{w} = \frac{Max - \min}{Max + \min} \qquad (2.3)$$

2.4.11. Perioada de oscilației.

O altă performanță a sistemului automat o constitue perioada T_0 de umplere şi golire a rezervorului. Evident, cu cât eroarea e este mai mică cu atât valoarea y a nivelului este mai constantă şi mai apropiată de referința w, dar manevrele de umplere-golire sunt mai repetate, adică T_0 este mic. În multe situații se alege eroarea e astfel încât $T_0 > 10$ minute.

2.4.12. Semnalizarea.

În afară de menținerea constantă a nivelului în rezervor[28] automatele semnalizează operatorului şi apariția diferitor evenimente dorite şi nedorite în funcționarea instalației. În acest exemplu simplu se semnalizează acțiunea de umplere a rezervorului prin intermediul lămpii H1 din circuitul 50 al schemei din Fig. 2.8.

2.4.13. Rolul operatorului.

În sfârşit, se observă ca la sistemele cu automat cu contacte şi releu din Fig. 2.7 şi Fig. 2.10 operatorul este totuşi prezent. Rolul său s-a redus însă numai la fixarea nivelelor de referință Max şi min. Din păcate schimbarea nivelului de referință constă în montarea electrozilor traductorului pe rezervor la distanțe diferite şi deci conducerea instalației de la distanță nu este posibilă în totalitate. Aceasta este o deficiență majoră a instalației care foloseşte pentru reglarea nivelului un automat cu contacte şi releu.

2.5. Sistem cu automat programabil logic - APL.

Automatul cu contacte şi releu din Fig. 2.7 şi Fig. 2.8 poate fi înlocuit foarte uşor, fără ca funcționarea şi performanțele să se modifice, cu un APL – Automat Programabil Logic[29].

2.5.1. Schema tehnologică cu echipamentul de automatizare.

În noua schemă tehnologică cu echipamentul de automatizare din *Fig. 2.11* automatul programabil logic apare reprezentat sub forma unui hexagon. Schimbarea pare minoră, în realitate blocul cu numărul 2 nu mai cuprinde un releu şi câteva contacte ca în Fig. 2.7, ci un adevărat microcalculator specializat pentru operarea cu variabile logice. Spre deosebire de calculatoarele tip PC automatele programabile sunt proiectate şi construite să funcționeze

[28] În acest caz nivelul este menținut între două limite Max şi min.

[29] In engleză PLC – Programmable Logic Controller.

sigur într-un mediu ostil, plin de perturbații. Din această cauză automatele programabile logice sunt relativ scumpe.

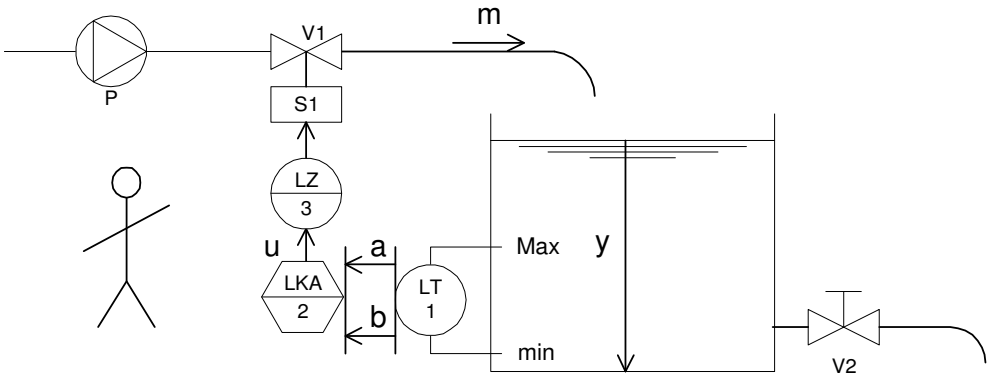


Fig. 2.11 Sistem cu automat programabil logic.

Din această cauză se consideră că este rentabil să înlocuieşti cu un APL un automat cu contacte şi relee numai dacă acest conține mai multe relee[30].

2.5.2. Schema desfăşurată electrică.

Schema desfăşurată electrică simplificată a sistemului[31] cu APL este prezentată în Fig. 2.12. Față de varianta cu contacte şi releu din Fig. 2.8 modificările sunt minime şi sunt efectuate numai din motive constructive. Astfel APL are intrări şi ieşiri pentru semnale tip contact. Deci la ieşirea sa apare un singur contact. Dar acest semnal trebuie să acționeze atât amplificatorul de putere format din contactorul K1 de la circuitul 30, cât şi lampa de semnalizare H1 din circuitul 50. Există mai multe soluții. Se poate folosi un releu intermediar cu singura funcțiune de multiplicator de contacte. Acesta ar transforma contactul k din circuitul 20 în alte două contacte care ar pute fi folosite în circuitele 30 şi 50. O alt soluție ar consta în folosirea a două variabile de ieşire a APL care iau întotdeauna aceiaşi valoare logică egală cu k. În sfârşit, se poate semnaliza, aşa cum s-a făcut în *Fig. 2.12*, acționarea contactorului K1 care comandă electromagnetul S1 şi care la rândul său deschide robinetul de umplere a rezervorului.

Automatele programabile logice sunt folosite pe scară largă în prezent, în special în industrie şi instalațiile pentru clădiri mari. Unele sisteme de protecție împotriva incendiilor, sau a furtului sunt de fapt tot automate programabile logice specializate.

---

[30] Automatul din Fig. 2.8 conține numai un releu şi este evident că nu este avantajoasă înlocuirea lui cu un APL.

[31] Instalației.

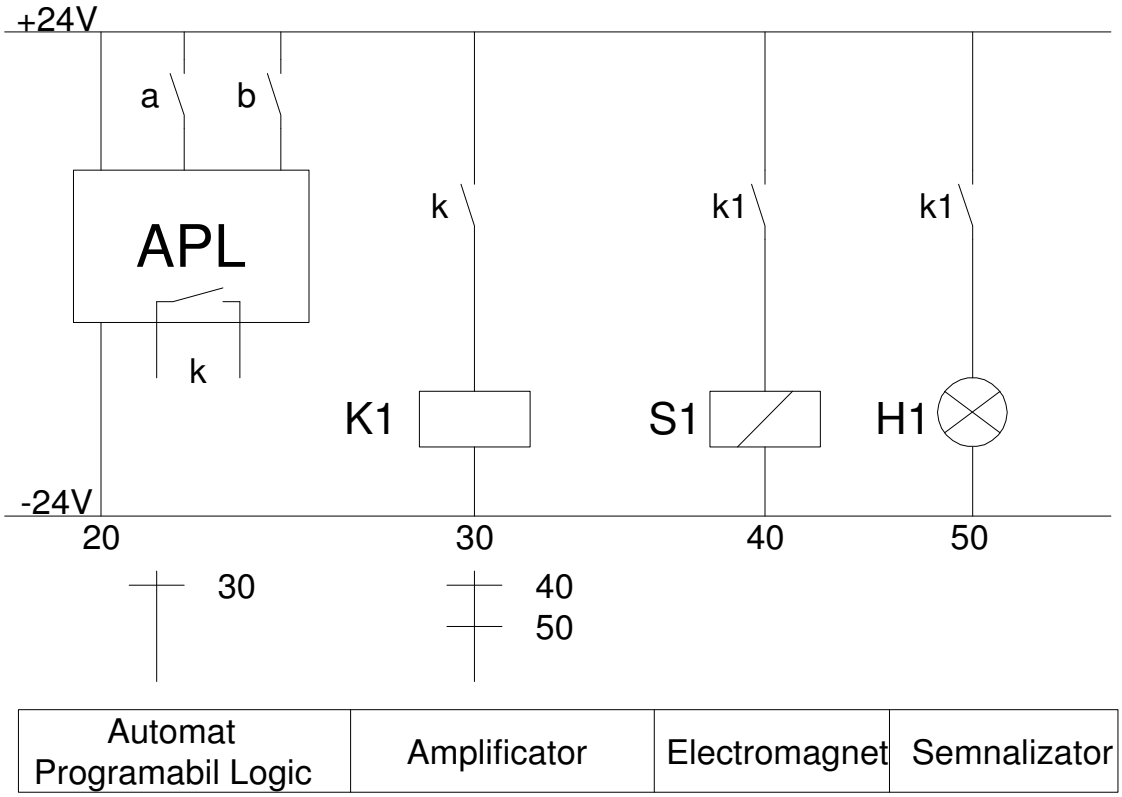


Fig. 2.12 Schema desfăşurată electrică simplificată pentru un sistem cu automat programabil logic

2.5.3. Programul.

Algoritmul unui sistem cu APL este prezentat sub formă de program. Proiectarea automatului programabil logic constă în acest caz, într-o mare măsură, în elaborarea programului APL. Această proiectare poate fi asistată de calculator care elaborează automat programul pe baza schemei desfăşurate electice echivalente bazate pe contacte şi relee. În cazul nostru trebuie să se introducă, eventual sub formă grafică, schema formată din circuitele 10 şi 20 din Fig. 2.8.

2.6. Sistem cu regulator bipoziţional - 2P.

O caracteristică tehnică importantă nouă a sistemului automat, stabilirea de la distanţă a valorii dorite a nivelului, numită referinţă, poate fi obţinută prin folosirea unui regulator bipoziţional.

2.6.1. Schema tehnologică cu echipamentul de automatizare.

Schema tehnologică a instalației în care apare acest regulatorul bipozițional este prezentată în Fig. 2.13. Traductorul de nivel 1 este format dintr-un element primar[32] care măsoară valoarea nivelului şi un transmițător la distanță a acestei valori[33]. De obicei transmiterea la distanță se face sub forma unui semnal unificat de curent cu valoarea cuprinsă între 4 şi 20 mA. Aparatul cu numărul 2 este regulatorul bipozițional[34]. El asigură sistemului aceleaşi performanțe determinate de relațiile (2.1), (2.2) şi (2.3), dar tehnologia de implementare a algoritmului bipozițional este diferită. Regulatorul este de obicei un aparat electronic. Semnalul de intrare în regulator este un semnal unificat de curent iar semnalul de ieşire este un contact k care acționează contactorul K1 al electromagnetului S1. din *Fig. 2.12*[35].

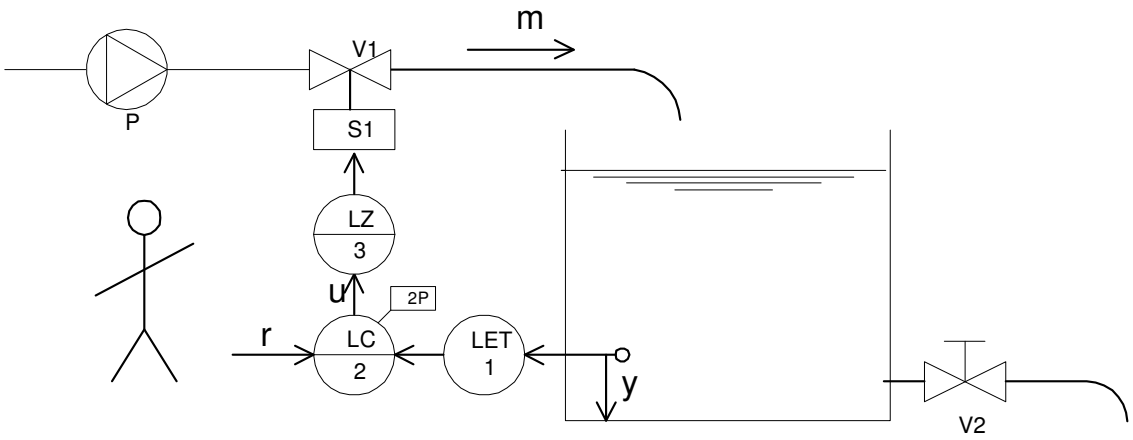


Fig. 2.13 Sistem automat cu regulator bipozițional (tip 2P) de nivel.

[32] Simbolul aparatului conține în poziția a doua litera E.

[33] Litera T din simbol.

[34] Litera C în poziția a doua şi un steag cu simbolul 2P.

[35] În această figură automatul programabil logic APL din circuitul 20 trebuie înlocuit cu regulatorul bipozițional.

2.6.2. Schema bloc

Schema bloc a unui sistem automat cu regulator bipoziţional este prezentată în *Fig. 2.14*. Faţă de *Fig. 2.9* se observă că a apărut regulatorul bipoziţional. Acesta este format dintr-un filtru - convertor, un comparator, desenat sub forma unui cerc şi un compensator al erorii.

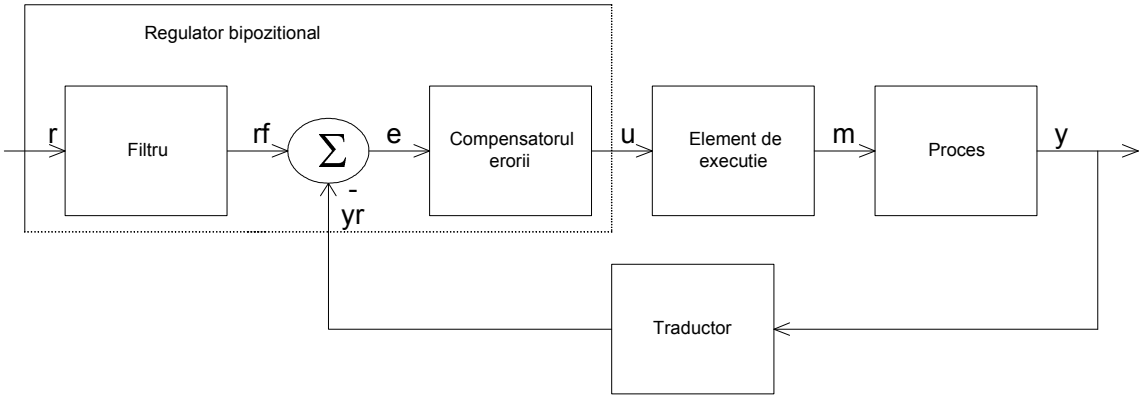


Fig. 2.14 Schema bloc a sistemului automat cu regulator bipoziţional.

2.6.3. Eroarea maximă.

Filtrul – convertor al regulatorului bipoziţional transformă semnalul introdus de operator, referinţa r, într-un semnal r_f. care să poată fi comparat cu mărimea de reacţie y_r. Blocul comparator al regulatorului realizează scăderea dintre semnalul r_f şi mărimea de reacţie y_r şi furnizează la ieşire eroarea e.

$$e = r_f - y_r \tag{2.4}$$

Semnalele r_f şi y_r trebuie să aibă aceiaşi dimensiune pentru ca să poată fi scăzute în relaţia (2.4). Ele sunt, de exemplu, semnale unificate de tensiune cu valoarea cuprinsă între 0 şi 10 volţi. Semnalul de referinţă r este introdus de operator, de exemplu sub forma unghiului de rotire a unui potenţiometru şi are altă unitate de măsură decât y_r.

Compensatorul furnizează la ieşire un semnal de comandă sub formă de contact u astfel încât intrarea sa, eroarea e, să tindă către zero. Relaţia intrare-ieşire a compensatorului este o curbă de tip ciclu histerezis cu forma din Fig. 2.15 în care

$$h = \frac{Max - \min}{2} \tag{2.5}$$

Comparând această relație cu relația (2.2) rezultă că în cazul sistemului automat cu regulator bipozițional eroarea maximă este:

$$e_{\max} = h \tag{2.6}$$

2.6.4. Legătura inversă negativă.

O caracteristică esențială a oricărui sistem automat[36] o constitue *legătura inversă negativă*[37] de la ieşirea procesului la intrarea regulatorului. Cuvântul negativ sugerează că semnalul y_r care circulă pe această cale se opune referinței r_f. Acest fenomen este subliniat prin indicarea semnului minus la intrarea comparatorului din *Fig. 2.14*.

2.6.5. Algoritmul bipozițional - 2P.

Algoritmul de funcționare al regulatorului bipozițional este acelaşi cu cel prezentat în Lista 2.1 şi folosit de automatul cu contacte şi releu sau automatul logic programabil. Modul lui de prezentare, Fig. 2.15, este însă mai puțin intuitiv pentru că se referă la eroarea e şi nu la valoarea nivelului y. Dacă eroarea este mare şi negativă[38], atunci mărimea de comandă a regulatorului sub forma contactului k ia valoarea 1, robinetul V1 se deschide şi rezervorul se umple. Dacă eroarea este mare şi pozitivă[39] atunci k = 0, robinetul V1 se închide şi rezervorul se goleşte. Pentru erori mici, cu valoarea situată în interiorul ciclului histerezis din Fig. 2.15, ieşirea k a compensatorului îşi păstrează valoarea precedentă. Acest fapt este indicat de săgețile prezente pe curba histerezis.

2.7. Sistem cu regulator proporțional - P.

Sistemele automate precedente au un algoritm de conducere bipozițional bazat pe un algoritm manual rudimentar de conducere. Conforma acestuia, robinetul V1 este deschis sau închis complet în funcție de semnul erorii, negativ sau pozitiv.

---

[36] Instalație automată.

[37] Reacția negativă.

[38] Adică e << h şi conform relației (2.4) r << y_r +h. Nivelul în rezervor este mult mai mare decât nivelul dorit determinat de referința r.

[39] Adică e >> h şi conform relației (2.4) r_f >> y_r +h. Nivelul în rezervor este mult mai mic decât nivelul dorit determinat de referința r_f.

2.7.1. Algoritmul proporțional - P.

O abordare mai rațională pentru conducerea sistemului ar consta în deschiderea robinetului V1 proporțional cu valoarea erori. Dacă eroarea este zero comanda este zero şi robinetul este închis. Dacă eroarea este mică comanda este mică. Cu cât valoarea erorii creşte cu atât comanda este mai mare. Deci relația intrare-ieşire pentru compensator este:

$$u = K_R e \tag{2.7}$$

în care K_R este constanta de proporționalitate a compensatorului regulatorului.

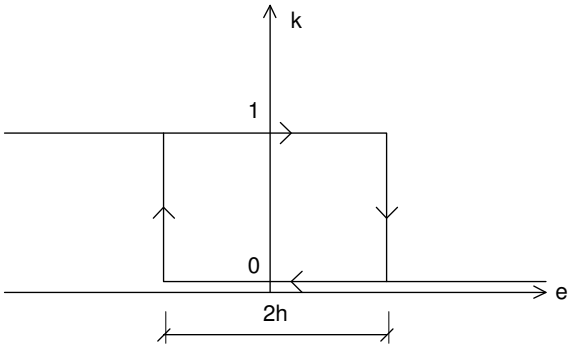


Fig. 2.15 Caracteristica tip histerezis a compensatorului regulatorului bipozițional.

2.7.2. Constanta de proporționalitate a compensatorului

Dacă se cunoaşte intrarea e în compensator şi ieşirea sa u se poate determina constanta lui de proporționalitate[40], numită adeseori constanta de proporționalitate a regulatorului:

$$K_R = \frac{u}{e} \tag{2.8}$$

Acțiunea compensatorului trebuie să fie aleasă constructiv în aşa fel încât comanda sa să *micşoreze* eroarea. În cazul de față eroarea e > 0 înseamnă că $r_f > y_r$, deci referința este mai mare decât ieşirea (nivelul) şi comanda u trebuie să închidă robinetul[41]. In felul acesta se asigură ***reacția negativă*** strict necesară funcționării sistemului automat.

---

[40] Aceasta poate fi interpretată drept panta dreptei care trece prin origine şi este determinată de relația (2.7).

[41] Deoarece reglarea nivelului se face prin umplerea rezervorului. Dacă reglarea s-ar fi făcut prin golire cu ajutorul robinetului V2, atunci semnalul u ar fi trebuit să deschidă robinetul. Regulatoarele au un comutator

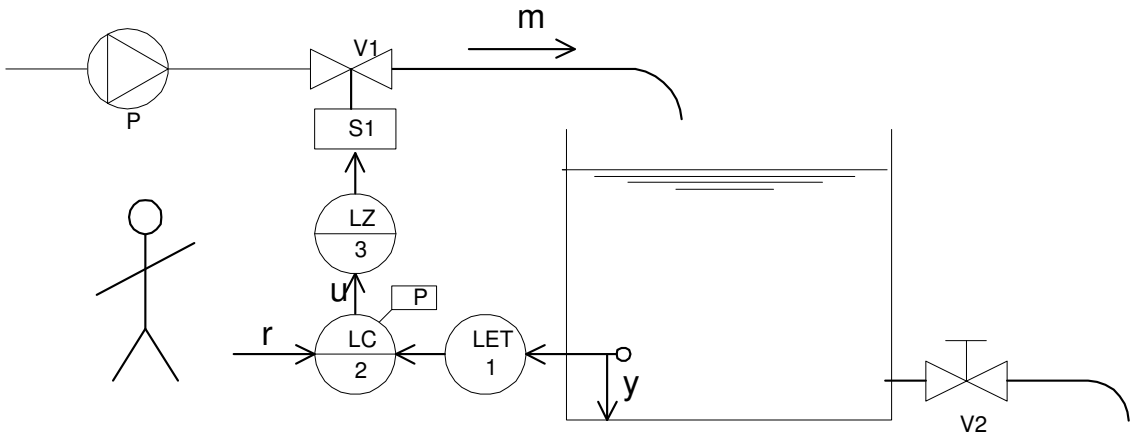


Fig. 2.16 Sistem automat cu regulator proporţional (tip P) de nivel.

2.7.3. Schema tehnologică cu echipamentul de automatizare.

Schema tehnologică cu echipamentul de automatizare a sistemului automat cu regulator proporţional din Fig. 2.16 este aproape identică cu cea din Fig. 2.13. Sunt vizibile următoarele două modificări. În primul rând ieşirea regulatorului nu mai este un semnal de tip contact k ci un semnal unificat[42] u cu valoarea dată de relaţia (2.7). În al doilea rând steagul ataşat aparatului cu numărul 2 trebuie să indice algoritmul de tip P (proporţional) şi nu 2P (bipoziţional). Aparatele 2 şi 3 au o construcţie diferită de cea a aparatelor din Fig. 2.13.

2.7.4. Schema bloc.

Schema bloc pentru sistemul automat cu regulator proporţional din *Fig. 2.17* este de asemenea foarte asemănătoare cu cea din *Fig. 2.14*. Bineînţeles că compensatorul regulatorului este proporţional iar ieşirea regulatorului este semnalul unificat u determinat de relaţia (2.7). Fiecare bloc al schemei poate fi caracterizat de o constantă de proporţionalitate K asemănătoare cu cea a compensatorului proporţional al regulatorului determinată de relaţia (2.8). Relaţia de definiţie a acesteia[43] este:

---

care permit operatorului ca în funcţie de modul de reglare (în cazul de faţă umplere – golire) să determine acţiunea corectă a compensatorului.

[42] Semnalul tip contact ia valori logice, de exemplu 0 sau 1, iar semnalul unificat este continuu, de exemplu o tensiune de 0..10V sau un curent de 4..20mA.

[43] Constanta K poate fi interpretată drept panta unei drepte care nu trece prin ogigine.

$$K = \lim_{t \to \infty} \frac{\Delta x_e}{\Delta x_i} \qquad (2.9)$$

în care Δx_i este o variație mică a semnalului de intrare în bloc,

Δx_e – o variație mică a semnalului de ieşire din bloc.

Definiția dată de relația (2.9) se calculează în regim staționar, adică după un timp foarte lung de la aplicarea variației semnalului de intrare. Aceasta este semnificația limitei pentru $t \to \infty$ care apare în relația (2.9).

2.7.5. Intrarea treaptă şi răspunsul indicial.

Cele mai simple şi mai frecvente manevre pe care le poate face operatorul instalației automatizate din Fig. 2.16 este să modifice brusc, sub formă de *treaptă*, referința r pentru a obține o nouă valoare a nivelului în rezervor. Dacă variația referinței este egală cu unitatea[44] atunci intrarea în sistem se numeşte ***treaptă unitară***[45]**.** Ieşirea unui sistem pentru o intrare treaptă unitară se numeşte ***răspuns indicial***.

2.7.6. Regimul tranzitoriu şi regimul staționar.

În momentul în care operatorul modifică referința r a sistemului automat din *Fig. 2.17*, semnalul se propagă spre ieşirea y, în sensul indicat de săgeți, cu anumite întârzieri şi oscilații datorate timpilor de transport, acumulărilor de energie şi material din sistem şi legăturii inverse negative. De exemplu, dacă în Fig. 2.16 operatorul modifică valoarea r a nivelului dorit în rezervor sistemul automat începe să funcționeze şi încearcă să aducă nivelul y la valoarea dorită. Dar aceasta nu se întâmplă instantaneu. Funcționarea sistemului automat în timp poate fi aproximativ următoarea. La început regulatorul deschide puțin robinetul V1. Apa este transportată de către pompa P în rezervor într-un timp finit. Durează şi acumularea apei în rezervor. Când s-a atins valoarea dorită a nivelului regulatorul închide puțin robinetul V1. Dar pe conductă mai este apă şi nivelul continuă să crească. Regulatorul închide mai mult robinetul şi nivelul începe să scadă, de data aceasta prea mult. Se deschide robinetul şi nivelul creşte iarăşi. Au loc, eventual[46], câteva oscilații până când nivelul se stabilizează la o anumită valoare. Deci după un timp suficient de mare creşterea treptată, uneori cu oscilații, a nivelului

---

[44] 1 m în cazul nostru.

[45] In funcție de sistemul de măsură ales *unitatea* poate lua practic orice valoare.

[46] Numai dacă distamța de la robinetul de reglare V1 la rezervor este foarte mare şi apare un timp de transport semnificativ.

se opreşte şi spunem că s-a terminat regimul tranzitoriu şi s-a intrat în regimul staţionar de funcţionare a sistemului automat. *Durata regimului tranzitoriu* este o performanţă importantă a sistemului automat.

2.7.7. Răspunsul indicial în regim staţionar.

Considerăm în continuare că se definesc semnalele de intrare şi ieşire ale blocurilor din *Fig. 2.17* astfel încât toate dreptele definite de relaţia (2.9) trec prin origine[47]. În aceste condiţii rezultă din relaţia (2.9) de definiţie a constantei de proporţionalitate că:

$$x_e = K \cdot x_i \tag{2.10}$$

Pentru un semnal de referinţă r tip treaptă unitară, în regim staţionar, fiecare bloc din schema *Fig. 2.17* este caracterizat de o constantă de proporţionalitate şi se poate deduce o expresie care stabileşte legătura între intrarea sistemului automat şi ieşirea sa y numit *răspuns indicial*.

$$y = K_p m = K_p K_e u = K_p K_e K_r e \tag{2.11}$$

$$e = r_f - y_r = K_c r - K_t y \tag{2.12}$$

$$y = \frac{K_d}{1 + K_d K_t} K_c \cdot r \tag{2.13}$$

în care

$$K_d = K_p K_e K_r \tag{2.14}$$

este constanta de proporţionalitate pe calea directă a buclei din *Fig. 2.17*.

[47] De exemplu alegând unităţi de măsură corespunzătoare.

2.7.8. Eroarea indicială în regim staționar.

Din relațiile (2.12) şi (2.13) se poate calcula o performanță importantă: eroarea staționară pentru o intrare tip treaptă a sistemului automat cu regulator proporțional:

$$e = \frac{1}{1 + K_d K_t} K_c \cdot r \qquad (2.15)$$

2.7.9. Exemplu numeric.

Să considerăm un exemplu numeric. Un rezervor cu înălțimea de 10m face parte dintr-o instalație cu regulator proporțional pentru reglarea nivelului. Intrarea (referința) r şi ieşirea (nivelul) y iau atunci valori cuprinse între 0 şi 10m. Semnalele r_f , y_r şi u din sistemul automat descris de schema bloc din *Fig. 2.17* sunt semnale unificate de tensiune cu valoarea cuprinsă între 1 şi 10V. Dacă K_r = 2 iar restul constantelor de proporționalitate sunt egale cu unu, atunci relațiile (2.13) şi (2.15) devin:

$$y = \frac{2}{3} r \qquad (2.16)$$

$$e = \frac{1}{3} r \qquad (2.17)$$

Deci dacă operatorul doreşte un nivel de 6m în rezervorul instalației şi modifică referința r sub formă de treaptă astfel încât să ia valoarea de 6m atunci regulatorul proporțional reglează automat nivelul la valoarea[48] y = 4m. Eroarea staționară în aceleaşi condiții este e = 2m. Evident că o astfel de performanță a sistemului automat este inacceptabilă. Ea poate fi îmbunătățită crescând constanta de proporționalitate a regulatorului. Dacă K_r=11 atunci pentru o referință[49] de 6m nivelul în rezervor va fi de 5,5m. În acest caz eroarea staționară s-a micşorat la 0,5.

[48] Acesta este răspunsul indicial în regim staționar.

[49] Valoare dorită a nivelului.

2.7.10. Stabilitatea şi robustețea.

Din păcate maniera de a reduce eroarea staționară prin creşterea constantei de proporționalitate poate face sistemul instabil[50]. *Un sistem este stabil dacă pentru orice intrare limitată are o ieşire limitată.* ***Stabilitatea*** sistemului automat este cea mai importantă caracteristică a sa. Se poate spune că o parte importantă a proiectării funcționale constă în asigurarea ***robusteței*** sistemului automat, adică a stabilității sale în condițiile existenței unui anumit grad de incertitudine referitor la caracteristicile componentelor sale.

2.7.11. Banda de proporționalitate a compensatorului.

Rezultatele obținute până acum pentru sistemul automat cu regulator tip P (proporțional) se bazează pe relația (2.10) care reprezintă o linie dreaptă. Din păcate în realitate regulatorul proporțional nu este liniar şi aceasta din cauză că intrarea şi ieşirea compensatorului erorii sunt semnale unificate cu un domeniu de variație a semnalului finit, de exemplu 0..10V..

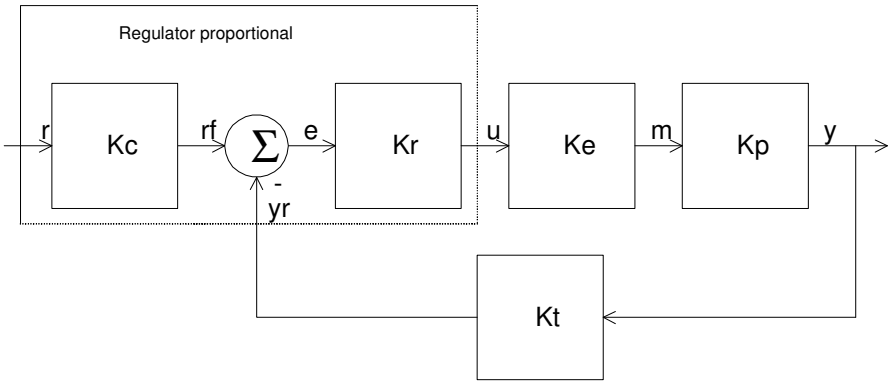


Fig. 2.17 Schema bloc a sistemului automat cu regulator proporțional.

În Fig. 2.18 este prezentată variația mărimii de comandă u în funcție de eroarea e pentru compensatorul regulatorului proporțional. Se observă că în afara domeniului 0..10V semnalele de intrare şi ieşire sunt limitate provocând un caracter neliniar pentru regulator.

[50] De exemplu dacă robinetul de reglare V1 este situat la o distanță mare de rezervor.

Pentru a pune în evidență această neliniaritate practicienii folosesc în locul caracteristicii constanta de proporționalitate K_R definită de relația (2.8) caracteristica banda de proporționalitate BP care exprimă în procente din domeniul erorii zona de liniaritate a regulatorului. Cu notațiile din Fig. 2.18 se obține:

$$K_R = \frac{D_u}{e_1} \tag{2.18}$$

Din definiția benzii de proporționalitate rezultă:

$$BP = \frac{e_1}{D_e} \cdot 100 \tag{2.19}$$

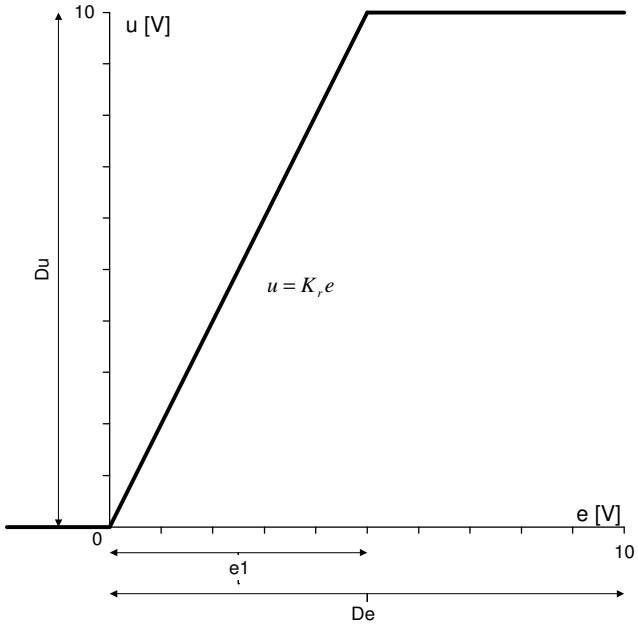


Fig. 2.18 Caracteristica statică a regulatorului proporțional.

Combinând aceste două relații se obține expresia benzii de proporționalitate BP în funcție de constanta de proporționalitate K_R:

$$BP = \frac{1}{K_R}\frac{D_u}{D_e}\cdot 100 \qquad (2.20)$$

Dacă eroarea e şi comanda u sunt semnale unificate de acelaşi tip atunci domeniile lor de variaţie D_e şi D_u sunt egale iar relaţia (2.20) devine:

$$BP = \frac{1}{K_R}\cdot 100 \qquad (2.21)$$

2.8. Sistem cu regulator direct.

Regulatoarele directe îşi iau energia necesară funcţionării din proces. Ele se numesc nivostate, termostate sau presostate dacă reglează nivelul, temperatura, respectiv presiunea.

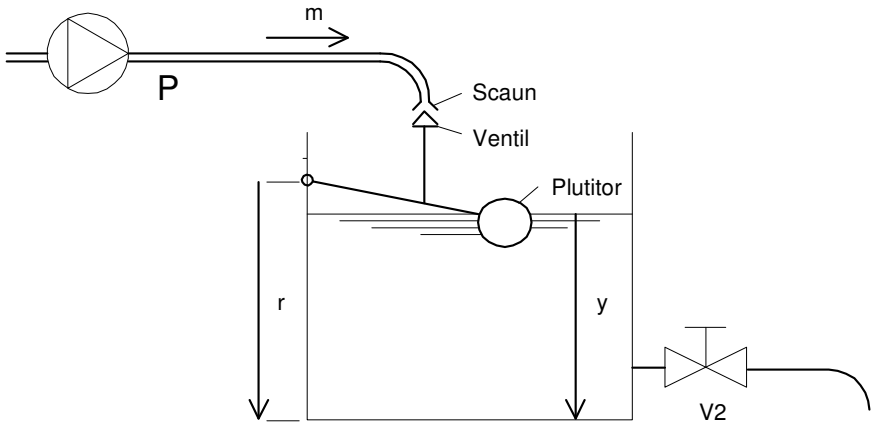


Fig. 2.19 Sistem cu regulator direct de nivel (nivostat).

În Fig. 2.19 se prezintă un rezervor al cărui nivel y este reglat cu ajutorul unui nivostat. Odată cu creşterea nivelului plutitorul se ridică, ventilul se apropie de scaun şi debitul de alimentare al rezervorului se micşorează. Dacă nivelul se micşorează are loc fenomenul invers. Plutitorul coboară, ventilul se depărtează de scaun şi debitul m creşte. În funcţie de forma ventilului şi scaunului algoritmul de reglare este de tip proporţional P sau bipoziţional 2P. Nivelul de referinţă r se stabileşte la valoarea dorită prin mutarea în sus sau în jos a articulaţiei plutitorului. Pentru alte tipuri de calculatoare referinţa poate fi stabilită de la

distanță, eventual cu ajutorul unui calculator. Eroarea e=r-y în regim staționar are valori apreciabile şi din această cauză nivostatele, ca şi celelalte tipuri de regulatoare directe, se folosesc numai în cazurile în care nu sunt necesare performanțe deosebite.

2.9. Sistem cu regulator proporțional integral derivativ – PID

Regulatoarele PID sunt cele mai folosite în practică atunci când se doresc performanțe bune pentru sistemul automat. Ele pot fi configurate astfel încât să funcționeze cu algoritm P, PI, PD sau PID.

2.9.1. Algoritmul PI.

Sistemul automat cu regulator tip P are întotdeauna o eroare staționară, după cum s-a arătat în paragraful 2.7.9. Anularea erorii staționare poate fi realizată cu ajutorul unui regulator proporțional integral (tip PI). Relația dintre intrarea şi ieşirea compensatorului erorii pentru acest regulator[51] este:

$$u(t) = K_r\left[e(t) + \frac{1}{T_i}\int e(t)dt\right] \qquad (2.22)$$

în care K_r este constanta de proporționalitate,

T_i – constanta de timp de integrare[52] măsurată în s/repetare.

Semnificația parametrilor K_r şi T_i ai regulatorului tip PI rezultă din răspunsul său la o intrare sub forma de treaptă cu amplitudinea unitară prezentat în Fig. 2.20. Se observă că durata T_i se măsoară la repetarea lui K_r.

[51] In practică se spune frecvent că această relație este a regulatorului şi nu a compensatorului.

[52] Constanta de timp pentru acțiunea de integrare, timpul de revenire (reset time).

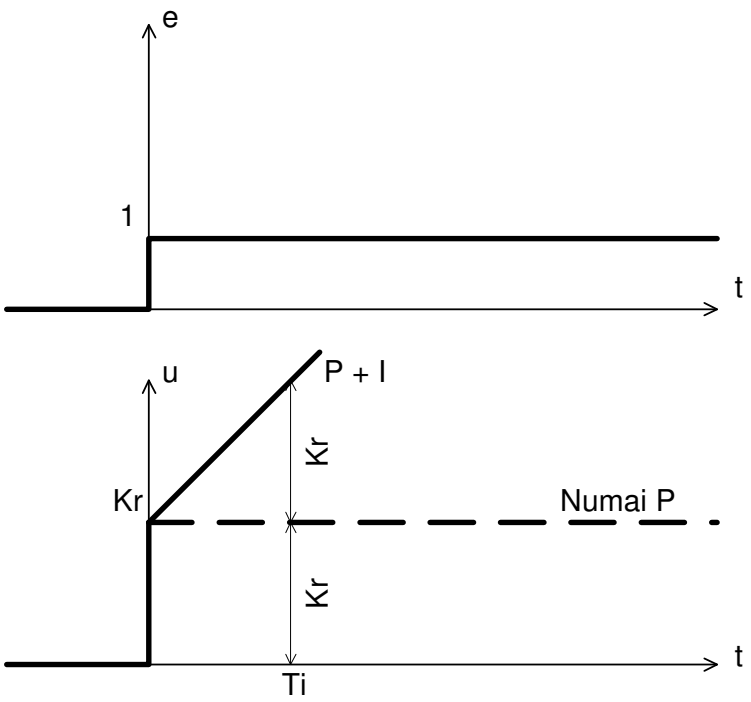


Fig. 2.20 Răspunsul regulatorului PI la o intrare treaptă.

Pentru a înțelege modul de funcționare al regulatorului tip PI să presupunem că la începutul funcționării sistemului automat componenta de acțiune de integrare este nulă. Pentru aceasta este suficient să atribuim constantei de timp de integrare T_i din (2.22) o valoare foarte mare ($T_i = \infty$). În funcție de valoarea constantei de proporționalitate K_r regulatorul amplifică eroarea rezultând o mărime de comandă care conduce în regim staționar la o mărime de ieşire y a sistemului automat diferită de valoarea dorită dată de mărimea de referință r. Relațiile (2.13) şi (2.15) pot fi folosite pentru calculul valorilor ieşirii y şi erorii e. Să atribuim acum o valoare finită constantei T_i. Deoarece în acest moment, fiind în regim staționar, există o eroare e constantă, comanda regulatorului u dată de (2.22) începe să crească la fel ca în Fig. 2.20. Datorită *reacției negative,* care se opune intrării, mărimea de comandă u se modifică până când eroarea devine zero. Din acest moment ieşirea regulatorului îşi păstrează valoarea atinsă. În realitate cele două acțiuni ale regulatorului, proporțională şi integrală, au loc simultan şi nu se petrec secvențial, aşa cum am prezentat mai înainte pentru o înțelegere mai uşoară. Efectul este însă acelaşi*, eroare staționară zero*. Orice îmbunătățire a unei performanțe se obține însă cu un anumit cost. În acest caz prețul plătit prin folosirea regulatorului PI constă în micşorare

robusteței sistemului automat şi implicit, creşterea duratei regimului tranzitoriu. Aceasta creştere apare datorită acțiunii suplimentare de integrare a regulatorului.

2.9.2. Algoritmul PD.

Algoritmul de reglare proporțional poate fi îmbunătățit impunând ca ieşirea u a compensatorului regulatorului să fie proporțională nu numai cu intrarea sa, eroarea e, ci şi cu viteza sa de variere. În felul acesta la apariția unor erori care se modifică rapid regulatorul acționează mai energic. Relația compensatorului tip PD este:

$$u = K_r(1 + T_d \frac{de}{dt}) \qquad (2.23)$$

în care K_r este constanta de proporționalitate,

T_d – constanta de timp de derivare[53] măsurată în s.

In Fig. 2.21 se observă efectul anticipativ al acțiunii derivative[54] şi efectul de proporționalitate cu viteza erorii[55]

2.9.3. Algoritmul PID.

O combinație între regulatorul PI şi regulatorul PD produce regulatorul PID ideal cu următoarea relație între intrarea şi ieşirea compensatorului:

$$u(t) = K_r(1 + \frac{1}{T_i}\int e(t)dt + T_d \frac{de(t)}{dt}) \qquad (2.24)$$

Din regulatorul PID ideal se poate obține regulatorul PI dacă T_d=0 sau regulatorul PD dacă T_i=∞.

In practică regulatoarele PID au algoritmi de funcționare uşor diferiți de (2.24). Motivele sunt variate[56]. Totuşi ideile de bază ale funcționării sunt aceleaşi cu cele expuse.

---

[53] Constanta de timp pentru acțiunea derivativă, timpul de anticipare, timpul vitezei erorii (rate time)

[54] Comparativ cu regulatorul tip P comanda regulatorului tip PD ia aceiaşi valoare cu T_d secunde mai devreme.

[55] De exemplu, la momentul t=0 comanda nu este zero deşi e=0.

[56] Acțiunea derivativă ideală din (2.24) nu poare fi realizată fizic. Uneori efectul derivativ se aplică ieşirii y şi nu erorii e. În sfârşit, unele firme preferă ca efectul derivativ real să fie să fie înmulțit şi nu adunat la algoritmul PI.

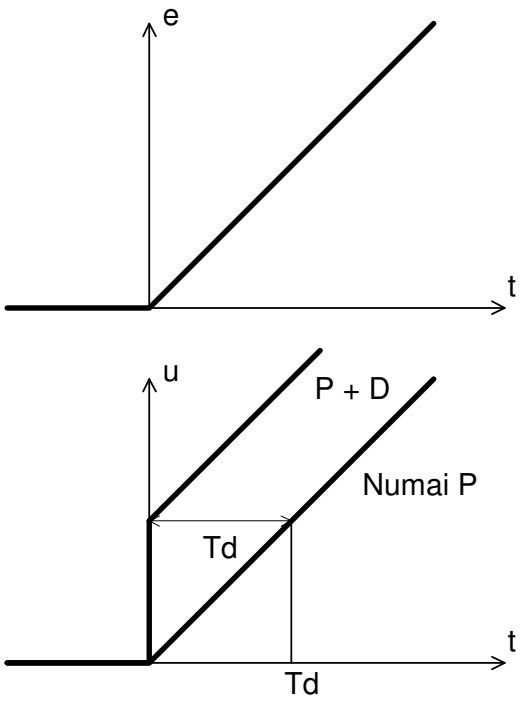


Fig. 2.21 Răspunsul regulatorului PD la o intrare rampă.

2.10. Sistem cu calculator.

Sistemele moderne sunt conduse de unul sau mai multe calculatoare[57] specializate[58] numite controlere, automate programabile[59] sau calculatoare de proces. În varianta cea mai simplă microcalculatorul este un automat sau regulator care realizează algoritmul de conducere. Acesta poate fi ales dintr-o gamă largă care are la bază algoritmii deja prezentați:

[57] Sistemele de conducere mai simple folosesc microcalculatoare.

[58] Spre deosebire de calculatoarele personale PC acestea sunt destinate să lucreze în medii neprietenoase, cu multe perturbații, iar condițiile de fiabilitate sunt mult mai severe. Deşi sunt realizate constructiv în mod diferit principiile de funcționare sunt în mare aceleaşi.

[59] Automatul programabil logic APL este un automat programabil AP particulare care lucrează numai cu semnale binare. Automatul programabil conține de multe ori două microcalculatoare specializate, unul calculează algoritmul şi altul realizează protocolul de comunicație.

automat secvențial[60], P, PI, PD şi PID. Folosirea calculatoarelor în conducerea proceselor pune însă probleme suplimentarea. Dintre acestea cea mai dificilă este legată de tipul finit de calcul a algoritmului de conducere. Cât timp se calculează algoritmul calculatorul nu mai primeşte informații despre starea sistemului şi nu trimite comenzi pentru conducerea sistemului. Din această cauză semnalele de intrare şi de ieşire din calculator sunt luate în considerare periodic numai la momente discrete de timp. Această operație se numeşte ***eşantionare***. Dacă eşantionarea se face cu o perioadă prea mare informațiile despre sistem sunt mai puține şi conducerea sa este mai proastă.

Folosirea calculatoarelor pentru conducerea automată a sistemelor este avantajoasă nu atât pentru calculul algoritmului, care am văzut că poate produce dificultăți datorită eşantionării, cât pentru noile funcțiuni pe care le pot îndeplini. Printre aceste menționăm: 1) Achiziția, prelucrarea şi memorarea unui volum imens de date, 2) Accesul rapid la datele memorate, 3) Realizarea unei interfețe grafice interactive şi prietenoase cu utilizatorul, 4) Instruirea operatorului şi ajutarea sa printr-un sistem expert la apariția unor defecțiuni, 5) Autotestarea sistemului automat şi autoacordarea regulatorului, 6) Autoacordarea automată adaptivă a regulatorului, 7) Combinarea mai multor sisteme conduse cu calculator într-o structură ierarhică, 8)Comunicarea la distanță cu alte calculatoare, 9) Transformarea traductoarelor şi elementelor de execuție în aparate inteligente. Aceste funcțiuni noi apar adeseori împreună şi creează aparate şi sisteme de conducere cu denumiri specifice. Câteva dintre ele sunt prezentate în continuare.

2.10.1. Conducerea de supraveghere şi achiziția datelor[61]

Una dintre caracteristicile cele mai importante ale unui sistem de conducere complex o constitue interfața cu operatorul. Posibilitățile grafice deosebite şi capacitatea de funcționare în mod interactiv a calculatoarelor moderne permit realizarea unor interfețe care să faciliteze la maximum activitatea operatorului. Punctul central al interfeței îl constitue schema sinoptică[62]. Un exemplu de schemă sinoptică este prezentat în Fig. 2.22.

---

[60] Automatul secvențial prezentat a fost de tip bipozițional 2P. Alte automate secvențial sunt folosite foarte frecvent, cum ar fi: automatul pentru pornirea directă a unui motor electric, automatul pentru pornirea stea – triunghi a unui motor, etc.

[61] SCADA – Supervisory Control And Data Acquisition.

[62] O schemă care permite observarea dintr-o privire a tuturor componentelor sistemului.

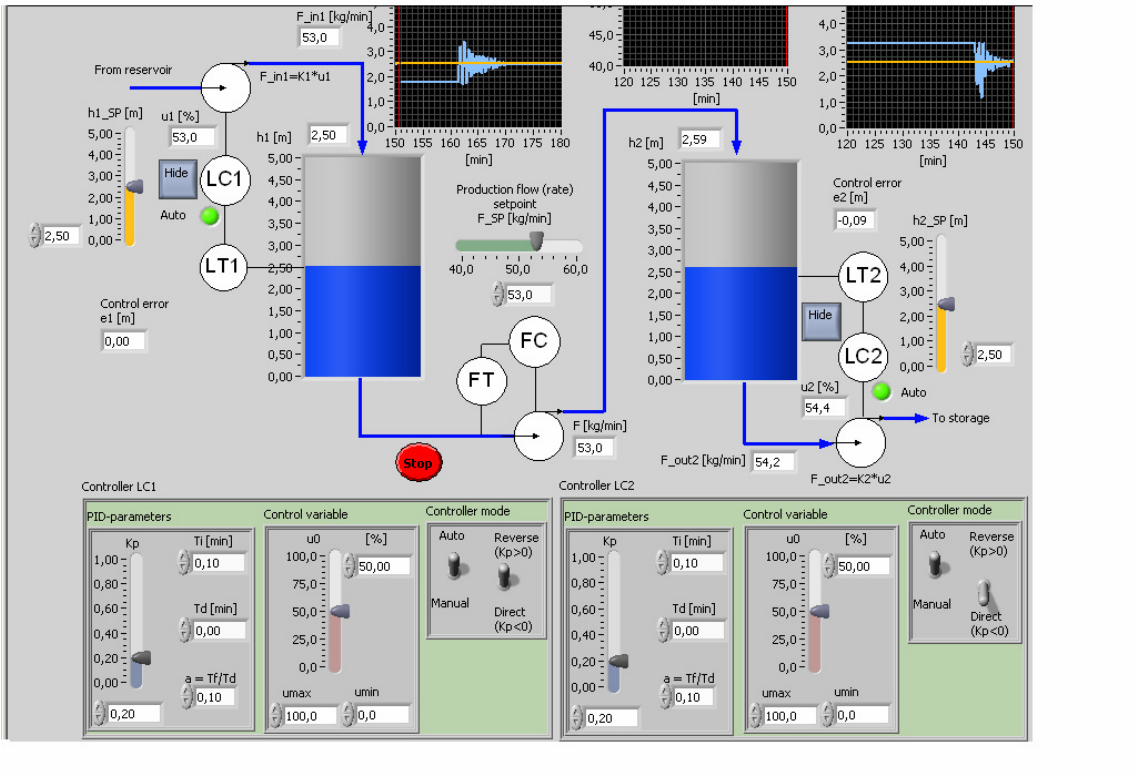


Fig. 2.22. Schema sinoptică a unui sistem de conducere cu calculator.

Obiectele din schema sinoptică permit, atunci când sunt selectate, afişarea unor noi scheme sinoptice sau a unor grafice şi tabele care prezintă în mod interactiv[63] starea sistemului pe baza informațiilor acumulate prin achiziția şi memorarea datelor furnizate de traductori. Chiar dacă operatorul nu întreprinde nici o acțiune, atunci când aceste informații ies dintr-un anumit domeniu se produc semnalizări şi alarme. În funcție de informațiile obținute operatorul poate realiza manual diferite operații de conducere. De exemplu poate porni sau opri anumite pompe cu ajutorul întrerupătoarelor prezentate în schema sinoptică.

[63] Funcționarea interactivă calculator – operator se realizează practic de cele mai multe ori în modul următor: operatorul selectează un obiect sau o acțiune prezentate în schema sinoptică iar calculatorul îi prezintă o serie de instrucțiuni şi operațiuni posibile în continuare.

2.10.2. Reglarea PID numerică.

Schema bloc a unui sistem automat cu regulator numeric PID este prezentată în Fig. 2.24. Comparativ cu *Fig. 2.17* s-a comasat elementul de execuție, procesul şi traductorul într-un bloc numit partea fixă a instalației[64]. Regulatorul numeric este format din comparator, două dispozitive de eşantionare cu perioada Te, compensatorul numeric al erorii realizat cu ajutorul unui calculator şi extrapolatorul de ordin zero[65] care transformă mărimea eşantionată într-o mărime continuă. Modul în care este eşantionată eroarea este prezentat în Fig. 2.24. Eşantioanele e_k sunt extrase la intervale egale cu perioada T_e şi sunt numerotate cu 0,1,2,3,…,k,…Valoarea lor poate fi prelucrată numeric de către calculator realizând algoritmul PID poziţional de conducere (2.24) sub forma:

$$u_k = K_R\left[e_k + \frac{1}{T_I}\sum_{n=0}^{n=k-1} e_n.T_e + T_D\,\frac{e_k - e_{k-1}}{T_e}\right] + u_0 \qquad (2.25)$$

în care u_0 este comanda inițială.

Se observă că integrala a fost înlocuită cu o sumă care aproximează suprafața acoperită de e(t) prin dreptunghiuri de lățime T_e ca în Fig. 2.24. Derivata este aproximată prin panta tangentei.

[64] Acest bloc este fix din punct de vedere al inginerului automatist care nu îl poate modifica.

[65] La ieşirea extrapolatorului valoarea este menținută constantă pe întreaga durată a perioadei de eşantionare.

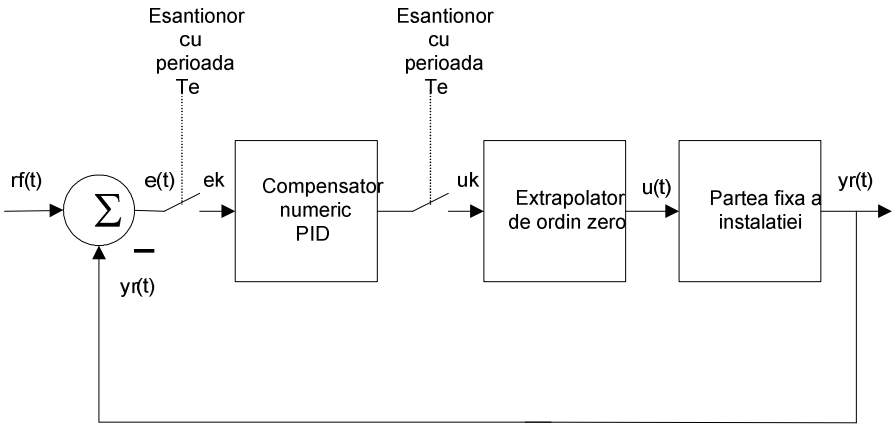


Fig. 2.23. Schema bloc a sistemului automat cu regulator numeric.

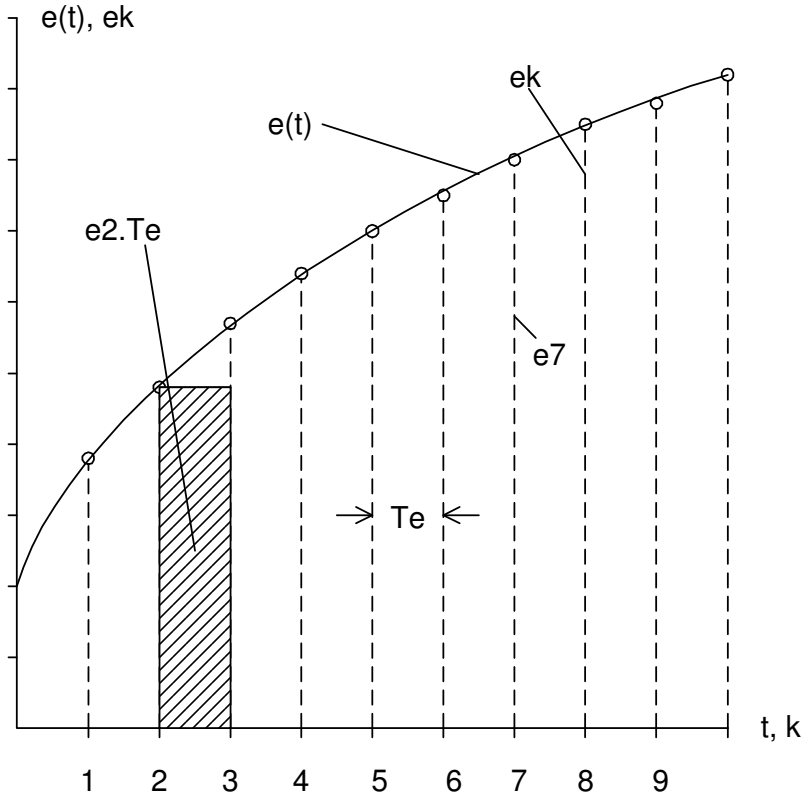


Fig. 2.24. Eşantionarea şi integrarea numerică a erorii e(t).

2.10.3. Autotestarea şi autoacordarea.

Înainte a începe să conducă sistemul şi ori de câte ori operatorul o cere, calculatorul se autotestează şi se autoacordează. Autotestarea se face asemănător cu cea efectuată de calculatoarele personale PC. În afară de acest tip de autotestare multe calculatoare de proces posedă un sistem[66] cu ceas de timp real separat care permite să se facă o autotestare continuă prin verificarea faptului că durata de executare a algoritmului de conducere nu a depăşit o anumită limită.

Autoacordarea realizează determinarea parametrilor algoritmului de reglare K_R, T_I, T_D şi T_e din (2.25) şi se face, în general, asemănător cu acordarea manuală a regulatorului.

2.10.4. Acordarea automată adaptivă.

Unele sisteme cu calculator permit acordarea continuă a regulatorului în timpul procesului de conducere. În felul acesta regulatorul se adaptează la eventualele schimbări al parametrilor procesului condus. În prezența unor perturbații puternice acordarea automată adaptivă durează mult şi nu produce, în general, rezultate prea bune.

2.10.5. Configurarea asistată de calculator.

Sistemele cu calculator prezintă şi avantajul că oferă foarte multe variante de conducere în funcție de echipamentul existent şi dorințele utilizatorului. Alegerea unei variante şi stabilirea configurației sistemului se poate face cu ajutorul calculatorului de conducere sau a unui calculator PC extern.

2.10.6. Sisteme de securitate la efracție şi siguranță la foc

Aceste sisteme specifice clădirilor sunt formate, în general, cu ajutorul unor automate programabile specializate. Unele traductoare ale acestor sisteme sunt inteligente în sensul că au cel puțin un sistem de adresare care să indice locul apariției evenimentului în clădire.

2.10.7. Sisteme domotice.

Dacă sistemul format din clădiri şi locuințe este condus de mai multe calculatoare se numeşte domotic. Aceste calculatoare sunt distribuite de obicei în spațiu la distanțe semnificative şi trebuie să se rezolve problema comunicațiilor dintre ele. O structură cu trei nivele ierarhice ca în Fig. 2.25 este frecvent folosită. Nivelul superior de conducere este de tip

[66] Cunoscut sub numele de watch dog.

supraveghere şi gestiune tehnică şi poate fi format dintr-o reţea locală LAN de calculatoare şi alte echipamente. Schemele sinoptice apar pe calculatoarele de la acest nivel. Tot aici sunt concentrate bazele de date şi facilităţile de obţinere a unor rapoarte tipărite şi de comunicare la mare distanţă prin radio sau telefon. De obicei supravegherea se face dintr-o cameră specială situată la o distanţă de instalaţiile distribuite în clădire care poate atinge până la 1000 metri.

Nivelul al doilea al sistemului ierarhic este ocupat de calculatoarele care realizează funcţiile de automatizare. De cele mai multe ori aceste calculatoare sunt de tipul automate programabile AP sau automate programabile specializate pentru securitate, control acces şi supraveghere şi alarmare la incendiu. Unele dintre aceste automate programabile pot fi conectate într-o reţea locală sau pot fi legate direct la calculatorul de supraveghere.

Al treilea nivel ierarhic este ocupat în Fig. 2.25 de traductoare şi elemente de execuţie. Acestea pot fi într-un număr foarte mare într-o clădire şi comunicarea dintre ele şi automatele programabile pun probleme legate atât de cablaj cât şi de protocol de comunicaţie. O soluţie modernă constă în folosirea unor aparate inteligente care conţin fiecare câte un microcalculator.

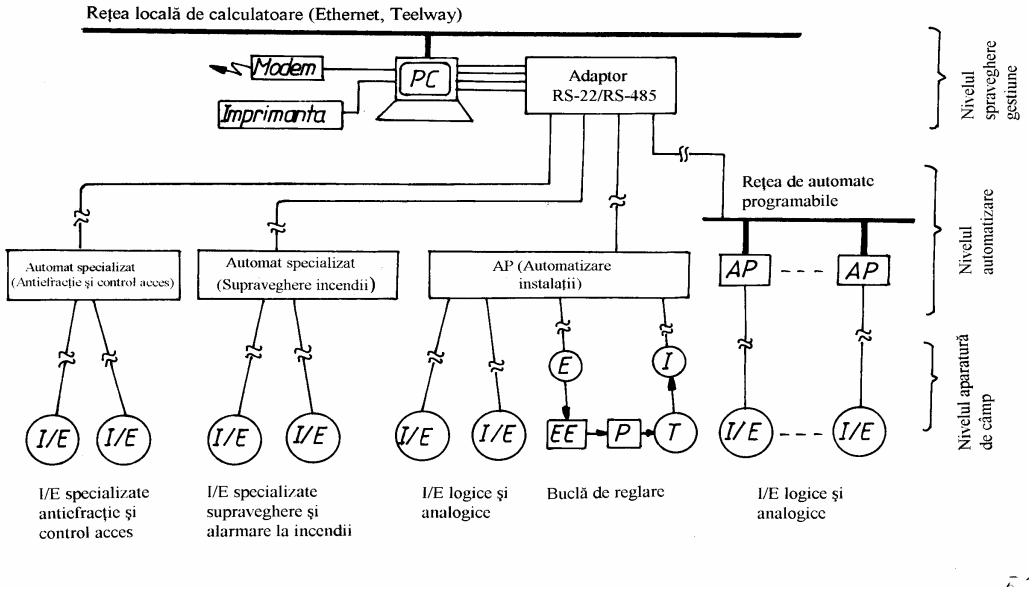


Fig. 2.25. Structura unui sistem automat ierarhic şi distribuit.

2.11. Modele cu variabile de stare ale sistemelor

Instalaţiile (sistemele de echipamente) sunt studiate cel mai adesea cu ajutorul sistemelor fizice cu parametrii concentraţi. In acest caz modelul matematic corespunzător este un sistem de ecuaţii diferenţiale. Modelul sistemic cel mai folosit este insa sistemul de ecuaţii

diferențiale de ordinul întâi, numit sistem de ecuații de stare. Obținerea ecuațiilor de stare este ilustrata printr-un exemplu tipic. In urma scrierii relațiilor fizice a componentelor unui sistem cu parametrii concentrați şi a aplicării condițiilor de echilibru şi de compatibilitate, se obține următoarea ecuație diferențiala:

$$\frac{d^3y}{dt^3} + a_2 \cdot \frac{d^2y}{dt^2} + a_1 \cdot \frac{dy}{dt} + a_0 \cdot y = c_2 \cdot \frac{d^2u}{dt^2} + c_1 \cdot \frac{du}{dt} + c_0 \cdot u \qquad (2.26)$$

in care

u(t) este excitația sistemului iar y(t) este răspunsul sistemului.

Se considera variabila intermediara z(t) conform următoarei scheme bloc:

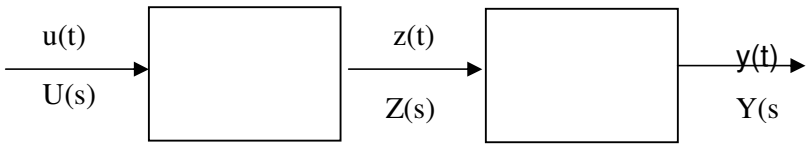


Fig. 2.26 Schema bloc a instalației cu explicitatea intrării u(t), ieşirii y(t) şi a variabilei intermediare z(t)

Din (2.26) şi *Fig. 2.26* rezulta:

$$\frac{Y(s)}{U(s)} = \frac{c_2 s^2 + c_1 s + c_0}{s^3 + a_2 s^2 + a_1 s + a_0} = \frac{Y(s)}{Z(s)} \cdot \frac{Z(s)}{U(s)} \qquad (2.27)$$

In (2.27) se pot face următoarele identificări:

$$\frac{Y(s)}{Z(s)} = c_2 s^2 + c_1 s + c_0 \qquad (2.28)$$

$$\frac{Z(s)}{U(s)} = \frac{1}{s^3 + a_2 s^2 + a_1 s + a_0} \qquad (2.29)$$

Trecând din nou in domeniul timp, relațiile (2.28) şi (2.29) devin:

$$y = c_2 \cdot \frac{d^2 z}{dt^2} + c_1 \cdot \frac{dz}{dt} + c_0 \cdot z \tag{2.30}$$

$$\frac{d^3 z}{dt^3} + a_2 \cdot \frac{d^2 z}{dt^2} + a_1 \cdot \frac{dz}{dt} + a_0 \cdot z = u \tag{2.31}$$

In (2.30) şi (2.31) se aleg variabilele de stare x_1, x_2 şi x_3 astfel:

$$x_1 = z \tag{2.32}$$

$$x_2 = \frac{dz}{dt} = \frac{dx_1}{dt} \tag{2.33}$$

$$x_3 = \frac{d^2 z}{dt^2} = \frac{dx_2}{dt} \tag{2.34}$$

Ecuația (2.34) poate fi pusa sub forma:

$$\frac{d^3 z}{dt^3} = \frac{dx_3}{dt} \tag{2.35}$$

Cu ajutorul relațiilor [(2.31)…(2.35)] se poate elimina variabila z şi se obține următorul sistem de ecuații de stare:

$$\begin{cases} \dfrac{dx_1}{dt} = x_2 & (2.36) \\ \dfrac{dx_2}{dt} = x_3 & (2.37) \\ \dfrac{dx_3}{dt} = -a_2 x_3 - a_1 x_2 - a_0 x_1 + u & (2.38) \end{cases}$$

Aceste ecuații pot fi scrise sub următoarea forma matriciala:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_2 & -a_1 & -a_0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot u \tag{2.39}$$

Procedând asemănător cu (2.30) se obține:

$$y = \begin{bmatrix} c_0 & c_1 & c_2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{2.40}$$

Sub forma prescurtata ecuațiile de stare (2.39) şi (2.40) se pot scrie in modul următor:

$$\begin{cases} \underline{\dot{x}}(t) = \underline{\underline{A}} \cdot \underline{x}(t) + \underline{b} \cdot u(t) & (2.41) \\ y(t) = \underline{c}^T \cdot \underline{x}(t) & (2.42) \end{cases}$$

unde

$\underline{\dot{x}}, \underline{x}, \underline{b}, \underline{c}$ *sunt vectori*

$\underline{\underline{A}}$ *este matrice*

u *este scalar.*

In cazul mai multor mărimi de intrare şi ieşire ecuațiile de stare au următoarea forma generala:

$$\begin{cases} \underline{\dot{x}}(t) = \underline{\underline{A}} \cdot \underline{x}(t) + \underline{\underline{B}} \cdot \underline{u}(t) & (2.43) \\ \underline{y}(t) = \underline{\underline{C}} \cdot \underline{x}(t) + \underline{\underline{D}} \cdot \underline{u}(t) & (2.44) \end{cases}$$

Cu ajutorul ecuațiilor de stare se poate determina răspunsul sistemului $\underline{y}(t)$ atunci când se cunosc intrarea sa $\underline{u}(t)$ şi condițiile inițiale $\underline{x}(0)$. Sa consideram cazul mai simplu al unui sistem cu o singura intrare u(t), o singura ieşire y(t) şi o singura stare x(t). Relația (2.43) se transforma intr-o ecuație diferențiala de ordinul întâi:

$$\dot{x}(t) = a \cdot x(t) + b \cdot u(t) \tag{2.45}$$

iar (2.44) se transforma intr-o ecuație algebrica de ordinul întâi:

$$y(t) = c \cdot x(t) + d \cdot u(t) \tag{2.46}$$

Relația (2.45) poate modela, de exemplu, variația nivelului x(t) intr-un rezervor. Acesta depinde atât de debitul de intrare u(t), cât şi de nivelul inițial din rezervor x(0). Dar nivelul inițial poate fi echivalat cu o intrare (sursă) suplimentară. Sistemul fiind liniar se poate aplica principiul superpoziției efectelor provocate de fiecare cauza separat:

$$x(t) = x_n(t) + x_f(t) \tag{2.47}$$

in care:

$x_n(t)$ este răspunsul natural al sistemului provocat numai de energia acumulata in elementele sistemului, deci pentru u(t)=0 (soluția generala a ecuației omogene, in termeni matematici);

$x_f(t)$ este răspunsul forțat al sistemului provocat numai de intrarea u(t) cu elementele sistemului relaxate, adică fără energii acumulate şi deci cu condiții inițiale nule x(0)=0 (soluția particulara a ecuației neomogene, in termeni matematici).

Din (2.45) rezulta ca răspunsul natural este soluția ecuației :

$$\dot{x}(t) = a \cdot x(t) \tag{2.48}$$

Se poate arata ca soluția cea mai generala este in acest caz:

$$x_n(t) = e^{at} \cdot K \tag{2.49}$$

in care K este o constanta care depinde de energia acumulata la momentul de timp inițial $t=t_0$ (condiția inițiala).

Dacă u(t)≠0 vom căuta o soluție generală a ecuației (2.45) de forma:

$$x(t) = e^{at} \cdot K(t) \tag{2.50}$$

in care K(t) este de data aceasta o funcție de timp.

Înlocuind soluția (2.50) in ecuația (2.45) rezulta după derivarea lui x(t):

$$\dot{x}(t) = a \cdot e^{at} \cdot K(t) + e^{at} \cdot \dot{K}(t) = a \cdot e^{at} \cdot K(t) + b \cdot u(t) \tag{2.51}$$

sau

$$\dot{K}(t) = e^{-at} \cdot b \cdot u(t) \tag{2.52}$$

Integrând ambele părți ale acestei relații rezulta:

$$K(t) = \int_{t_0}^{t} e^{-a\tau} \cdot b \cdot u(\tau) d\tau + K(t_0) \tag{2.53}$$

Înlocuind în (2.50) se obține:

$$x(t) = e^{at} \cdot \left[\int_{t_0}^{t} e^{-a\tau} \cdot b \cdot u(\tau) d\tau + K(t_0) \right] \tag{2.54}$$

Pentru determinarea lui $K(t_0)$ se face in (2.50) $t=t_0$:

$$K(t_0) = e^{-at_0} \cdot x(t_0) \tag{2.55}$$

şi deci soluția ecuației de stare (2.45) devine :

$$x(t) = e^{a(t-t_0)} \cdot x(t_0) + \int_{t_t}^{t} e^{a(t-\tau)} \cdot b \cdot u(\tau) d\tau \tag{2.56}$$

şi se poate distinge răspunsul natural $x_n(t)$ de cel forțat $x_f(t)$.
Prin analogie cu (2.45) se poate scrie soluția ecuației matriciale de stare (53):

$$\underline{x}(t) = e^{\underline{\underline{A}}(t-t_0)} \cdot \underline{x}(t_0) + \int_{t_0}^{t} e^{\underline{\underline{A}}(t-\tau)} \cdot \underline{\underline{B}} \cdot u(\tau) d\tau \tag{2.57}$$

in care exponențiala matriciala e^{At} se defineşte astfel:

$$\underline{\underline{\Phi}}(t) = e^{\underline{\underline{A}}t} = \underline{\underline{1}} + \underline{\underline{A}}t + \frac{1}{2!}(\underline{\underline{A}}t)^2 + \ldots + \frac{1}{n!}(\underline{\underline{A}}t)^n + \ldots \tag{2.58}$$

şi se numeşte matricea de tranziţie a stărilor sistemului.

Răspunsul natural al sistemului rezulta imediat din (2.57) daca intrarea $\underline{u}(t) = 0$ şi se cunoaşte starea iniţiala $\underline{x}(t_0)$:

$$\underline{x}(t) = e^{\underline{\underline{A}}(t-t_0)} \cdot \underline{x}(t_0) \tag{2.59}$$

Daca evaluarea variabilelor de stare se face la un interval constant de timp $T=t-t_0$ (70), atunci se poate calcula cu ajutorul (2.59):

$$\underline{x}(t_0 + T) = e^{\underline{\underline{A}}T} \cdot \underline{x}(t_0) \tag{2.60}$$

$$\underline{x}(t_0 + 2T) = e^{\underline{\underline{A}}T} \cdot \underline{x}(t_0 + T) \quad ...etc \tag{2.61}$$

in care *matricea de tranziţie a stărilor* $e^{\underline{\underline{A}}t}$ *se calculează o singura data cu ajutorul seriei (2.58).*

Se observa ca

$$\frac{1}{10!} = 0.276 \cdot 10^{-6}$$

Deci daca se iau zece termeni in aproximarea seriei se obţin elementele matricei de transfer cu 10 cifre semnificative exacte. Câteva cazuri particulare sunt importante. Daca sistemul are o singura variabila de stare, atunci ecuaţia de stare şi răspunsul natural devin:

$$\dot{x}(t) = a \cdot x(t) \tag{2.62}$$

$$x_n(t) = e^{a(t-t_0)} \cdot x(t_0) \tag{2.63}$$

Daca $t_0=0$ şi condiţia iniţiala $x(0)=1$ atunci răspunsul natural este o exponenţială:

$$x_n(t) = e^{at} \tag{2.64}$$

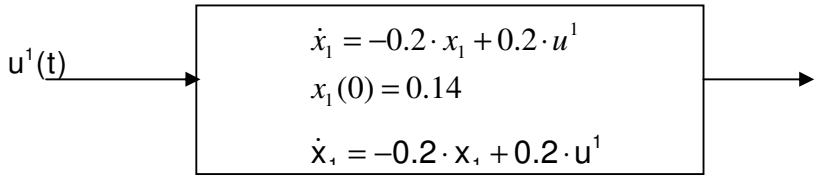
Pentru a=0 ecuaţia (2.62) devine

$$\dot{x}(t) = 0 \tag{2.65}$$

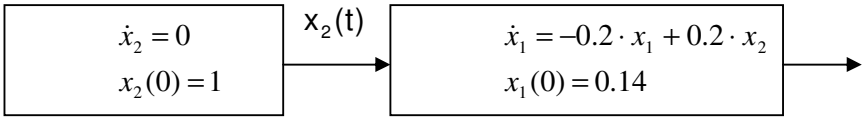
iar răspunsul natural este o treapta unitate

$$x(t) = u^1(t) \tag{2.66}$$

Folosind răspunsurile naturale ale unor sisteme particulare se poate transforma problema determinării *răspunsului general* (2.47) al unui sistem in problema determinării *răspunsului natural* al unui sistem lărgit.



(a)



(b)

Fig. 2.27 Transformarea unui răspuns forţat in răspunsul natural al unui sistem lărgit

In *Fig.2.27* se prezintă un exemplu. Sistemul este descris de o singura ecuație de stare de tipul (2.45) cu a=-0.2, b=0.2 şi condiția inițiala $x_1(0)=0.14$.Marimea de intrare este treapta unitate $u^1(t)$ care poate fi *generata* de ecuația (2.65).Introducând generatorul treptei unitate la intrarea in sistem (*Fig. 2.27*b) se obține un sistem mai mare, descris de doua ecuații de stare, dar *omogen*, fără intrare exogena (din afara sistemului):

$$\dot{x}_1 = -0.2 \cdot x_1 + 0.2 \cdot x_2$$
$$\dot{x}_1(0) = 0{,}14 \tag{2.67}$$
$$\dot{x}_2 = 0 \tag{2.68}$$
$$x_2(0) = 1$$

sau in notație matriciala:

$$\underline{\dot{x}} = \underline{\underline{A}} \cdot \underline{x}(t) \tag{2.69}$$

in care:

$$\underline{\underline{A}} = \begin{bmatrix} -0.2 & 0.2 \\ 0 & 0 \end{bmatrix} \qquad (2.70)$$

$$\underline{x}(0) = \begin{bmatrix} 0.14 \\ 1 \end{bmatrix}$$

Alegem momentul inițial de timp t_0=0 şi intervalul de evaluare a variabilelor T=0.5 s. Soluția ecuațiilor de stare poate fi evaluata cu ajutorul relațiilor (2.60) şi (2.61).

$$\underline{\underline{AT}} = \begin{bmatrix} -0.2 & 0.2 \\ 0 & 0 \end{bmatrix} \cdot 0.5 = \begin{bmatrix} -0.1 & 0.1 \\ 0 & 0 \end{bmatrix}, \ (\underline{\underline{AT}})^2 = \begin{bmatrix} 0.01 & -0.01 \\ 0 & 0 \end{bmatrix}, \qquad (2.71)$$

$$(\underline{\underline{AT}})^3 = \begin{bmatrix} -0.001 & 0.001 \\ 0 & 0 \end{bmatrix} \qquad (2.72)$$

$$e^{\underline{\underline{AT}}} \cong \underline{\underline{1}} + (\underline{\underline{AT}}) + \frac{1}{2!}(\underline{\underline{AT}})^2 + \frac{1}{3!}(\underline{\underline{AT}})^3 = \qquad (2.73)$$

$$= \begin{bmatrix} 1 - 0.1 + 0.005 - 0.00017 & 0.1 - 0.005 + 0.00017 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.90483 & 0.09517 \\ 0 & 1 \end{bmatrix}$$

Aplicând (2.60) rezulta:

$$\underline{x}(0.5) = \begin{bmatrix} 0.90483 & 0.09517 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0.14 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.22185 \\ 1 \end{bmatrix} \qquad (2.74)$$

Soluția exacta a ecuației de stare din *Fig. 2.27*a este următoarea:

$$x_1(t) = x_n(t) + x_f(t) = 0.14 \cdot e^{-\frac{t}{5}} + (1 - e^{-\frac{t}{5}}) \qquad (2.75)$$

Pentru t=0.5 s avem:

$$x_1(0.5) = 0.22184 \qquad (2.76)$$

adică aceeaşi cu soluția (2.74), cu excepția celei de-a 5-a cifre semnificative.

Cele patru modele – matematic, structural, funcțional şi cu variabile de stare – ale sistemului fizic pot fi folosite separat sau împreună. Alegerea soluției depinde foarte mult de tipul problemei, de experiența inginerească existentă şi de programele disponibile. Câteva medii de dezvoltare gratuite ale aplicațiilor pentru sisteme fizice sunt Dynast, KitSAS Matlab şi Scilab.

2.12. Modele în domeniul frecvență ale sistemelor

În domeniul frecvență sistemele pot fi analizate şi sintetizate cu ajutorul seriei Fourier şi transformărilor Fourier, Laplace şi Z.

Prelucrarea în domeniul frecvență se aplică în primul rând semnalelor care se modifică în timp dar variabila independentă poate fi şi alta, de exemplu spațiul. Metoda a fost fundamentată la început pentru sistemele fizice *liniare* şi se bazează pe proprietatea remarcabilă a acestora de *fidelitate sinusoidală*, adică pe faptul că răspunsul unui sistem liniar la un semnal sinusoidal este tot un semnal sinusoidal, cu aceiaşi frecvență dar cu amplitudine şi fază diferite.

În afară de studiul propagării semnalelor prin sisteme liniare, prelucrarea în domeniul frecvență se mai foloseşte la identificarea surselor de semnal, analiza şi sinteza sunetelor şi imaginilor şi depistarea defecțiunilor.

Aplicarea largă a acestor metode în practică a condus la fabricarea în serie a unor microprocesoare specializate[67] pentru acest gen de operații care sunt folosite într-un evantai larg de aplicații, de la telefoane mobile la instrumente ştiințifice.

[67] DSP – Digital Signal Processor.

Iată alte motive pentru care prelucrarea semnalelor în domeniul frecvență este importantă.

- Mişcarea naturală a sistemelor fizice provocată de energia acumulată este de formă sinusoidală sau exponențială.
- Sistemele fizice pot fi caracterizate prin frecvențele proprii.
- Prelucrarea numerică a semnalelor implică o analiză în domeniul frecvență pentru determinarea unei eşantionări corecte.
- Multe perturbații şi zgomote pot fi caracterizate prin frecvențe ridicate şi pot fi filtrate.
- Analiza semnalelor aleatoare se face în domeniul frecvență.
- Descompunerea semnalelor periodice în serie Fourier este ortogonală. Prin urmare determinarea unor noi componente nu necesită recalcularea componentelor vechi.
- Există algoritmul de transformare Fourier rapidă (FFT).
- Multe semnale din natură sunt sinusoidale, de exemplu curentul alternativ, vibrațiile şi sunetele.
- Analiza stabilității sistemelor fizice şi stabilitatea algoritmilor de prelucrare numerică se face în domeniul frecvență.
- Analiza în domeniul frecvență este mai uşoară pentru calculul manual. Chiar dacă analiza se face asistată de calculator, interfața cu utilizatorul foloseşte tot conceptele din domeniul frecvență.
- Multe performanțe şi limitări fizice sunt exprimate în domeniul frecvență. De exemplu realizabilitatea fizică a sistemului impune ca funcția sa de transfer să aibă gradul numărătorului mai mare sau egal cu gradul numitorului.

2.12.1. Scurt istoric

Analiza Fourier a fost preconizată pentru prima dată de Jean Baptiste Joseph Fourier (1768 – 1830), un matematician şi fizician francez. Fourier era interesat de propagarea căldurii şi a prezentat la Academia Franceză în 1807 un articol în care folosea sinusoidele la reprezentarea distribuțiilor de temperatură. Lucrarea afirma că orice semnal periodic poate fi reprezentat printr-o sumă de sinusoide alese corespunzător. Printre referenții lucrării se aflau şi doi matematicieni faimoşi: Josepf Louis Lagrange (1736 – 1813) şi Pierre Simon de Laplace (1749 – 1827).

Laplace şi alți referenți au votat pentru publicarea lucrării dar Lagrange a protestat vehement. În următorii ani Lagrange a insistat că o astfel de abordare nu poate fi folosită pentru reprezentarea semnalelor cu colțuri, adică cu forme discontinue aşa cum sunt formele de undă rectangulare. Academia Franceză s-a înclinat în fața prestigiului Lui Lagrange şi a respins lucrarea. Abia după 15 ani, după moartea lui Lagrange lucrarea a putut fi publicată. Din fericire Fourier a avut alte activități care i-au reținut atenția în această perioadă, politica, expediția lui Napoleon În Egipt şi evitarea ghilotinei din timpul Revoluției Franceze.

Lagrange a avut dreptate când a afirmat că prin însumarea unor sinusoide nu se pot sintetiza semnale cu colțuri. Totuşi se pot obține aproximări foarte bune. Eroarea care apare la analiza semnalelor datorită colțurilor este cunoscută în prezent sub numele de efect Gibbs.

2.12.2. Semnale sinusoidale. Funcția de transfer.

Există multe reprezentări ale semnalelor sinusoidale. O formă importantă este următoarea:

$$x(t) = A_k \cos(\omega_k t + \varphi_k) \quad , \quad -\infty < t < \infty \qquad (2.77)$$

În această relație A reprezintă amplitudinea, ω pulsația şi φ faza.

O observație importantă o constituie faptul că în acest caz sinusoidele sunt reprezentate de fapt prin cosinusoide şi acestea sunt definite de la minus infinit la plus infinit, diferind de reprezentarea tradițională în care sinusoida începe din zero cu valoarea zero.

În relația (2.77) pulsația poate apare în următoarele două forme diferite

$$\omega_k = 2\pi f_k = \frac{2\pi}{T_k} \qquad (2.78)$$

în care f este frecvența sinusoidei iar T este perioada sa.

Un rol important în prelucrarea datelor în domeniul frecvență îl joacă relația lui Euler:

$$e^{j\beta} = \cos\beta + j\sin\beta \qquad (2.79)$$

Pentru unghiuri negative,

$$e^{-j\beta} = \cos\beta - j\sin\beta \qquad (2.80)$$

Adunând şi scăzând aceste relații rezultă

$$\cos\beta = \frac{e^{j\beta} + e^{-j\beta}}{2} \qquad (2.81)$$

$$\sin\beta = \frac{e^{j\beta} - e^{-j\beta}}{2j} \qquad (2.82)$$

Înlocuind pe (2.81) în (2.77)) se obține forma complexă a semnalului sinusoidal de pulsație ω_k:

$$x(t) = A_k \frac{e^{j(\omega_k t+\varphi_k)} + e^{-j(\omega_k t+\varphi_k)}}{2} = X_k e^{j\omega_k t} + X_{-k} e^{-j\omega_k t} \qquad (2.83)$$

în care

$$X_k = \frac{A_k}{2} e^{j\varphi_k} \qquad (2.84)$$

$$X_{-k} = \frac{A_k}{2} e^{-j\varphi_k} \qquad (2.85)$$

2.12.3. Propagarea semnalelor sinusoidale prin sisteme liniare

Forma complexă (2.83) este potrivită pentru studiul propagării sinusoidelor prin sistemele liniare. Aplicând principiul superpoziției se poate studia propagarea celor două componente separat. Să considerăm prima componentă numită sinusoidă complexă:

$$x_1(t) = X_k e^{j\omega_k t} \qquad (2.86)$$

Datorită fidelității sinusoidale ieşirea sistemului liniar va avea aceiaşi pulsație ω_k dar o amplitudine şi fază diferite

$$y_1(t) = Y_k e^{j\omega_k t} \qquad (2.87)$$

Amplitudinea semnalului de ieşire este un număr complex de forma:

$$Y_k = \frac{B_k}{2} e^{j\psi_k} \qquad (2.88)$$

Se procedează la fel şi cu a doua componentă $x_2(t)$ şi se obține $y_2(t)$. Răspunsul sistemului va fi:

$$y(t) = y_1(t) + y_2(t) \qquad (2.89)$$

Sistemele fizice se comportă la trecerea semnalelor prin ele ca nişte filtre trece jos, adică atenuează puternic sinusoidele cu frecvențe înalte.

2.12.4. Funcția de transfer a filtrului analogic de ordin unu.

Se analizează efectul de filtru trece jos a sistemului pompă – rezervor. Acesta lasă să treacă prin el numai semnalele sinusoidale de frecvență joasă. La început se obține modelul matematic liniar.

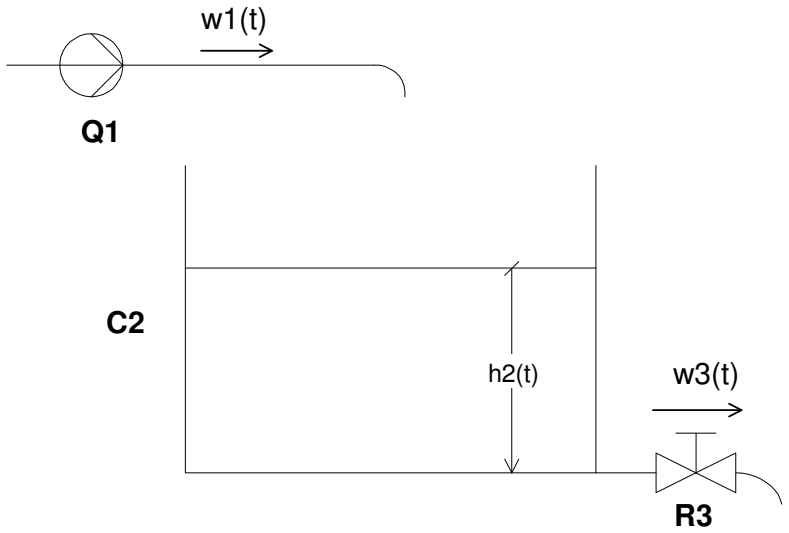


Fig. 2.28 Sistemul pompă rezervor

Aplicând legea conservării masei[68] la sistemul pompă - rezervor din *Fig. 2.28* rezultă următoarea ecuație diferențială neliniară:

$$\frac{d}{dt}(A\rho h_2) = w_1 - w_3 = w_1 - \alpha\sqrt{h_2} \qquad (2.90)$$

sau

$$\frac{dh_2}{dt} + c\sqrt{h_2} = bw_1 \qquad (2.91)$$

Considerând variații mici ale nivelului Δh_2 în jurul nivelului h_0 se obține:

$$\Delta(\frac{dh_2}{dt}) + c\Delta(\sqrt{h_2}) = b\Delta w_1 \qquad (2.92)$$

[68] Masa acumulată în rezervor este egală cu diferența dintre debitul masic de intrare şi debitul masic de ieşire.

Pentru variații foarte mici operatorul Δ poate fi înlocuit cu operatorul diferențial d.

$$d(\sqrt{h_2}) = \left.\frac{d\sqrt{h_2}}{dh_2}\right|_{h_2=h_0} dh_2 = \frac{1}{2\sqrt{h_0}} dh_2 \qquad (2.93)$$

Pentru simplificarea scrierii notăm $dh_2 = h$ şi $dw_1 = w$. Ecuația diferențială, de data această liniară, este:

$$\frac{dh}{dt} + \frac{c}{2\sqrt{h_0}} h = b.w \qquad (2.94)$$

Această ecuație este valabilă numai pentru variații mici ale nivelului h în jurul valorii inițiale h_0.

Modelul liniar (2.94) al sistemului se scrie sub forma:

$$T\frac{dy(t)}{dt} + y(t) = Kx(t) \qquad (2.95)$$

în care T este o constantă de timp măsurată în secunde iar K o constantă de proporționalitate.

Să considerăm exemplul numeric cu c=2 şi b=1. Se calculează semnalul de ieşire al sistemului, adică nivelul, *în regim staționar* pentru o intrare constantă. Un semnal constant de la minus infinit la plus infinit poate fi considerat că provine dintr-o sinusoidă (2.77) cu pulsația egală cu zero[69]. Se consideră următoarele situații:

1. Rezervorul din *Fig. 2.28* este gol, debitul constant w_1=12, modelul este neliniar. Din (2.91) rezultă h_{2st}=36.
2. Rezervorul umplut la nivelul h_0=36, debitul este constant şi mic w=1,2 iar modelul este liniar. Folosind relația (2.94) se obține h_{st}=7,2. Deci valoarea staționară a nivelului din rezervorul din *Fig. 2.28* va fi h_{2st}=43,2.
3. Daca încercăm să rezolvăm cazul precedent fără să folosim modelul liniarizat prin aproximare (2.94) al rezervorului vom introduce în modelul neliniar (2.91) valoarea debitului total w_1=12+1,2=13,2 şi rezultă h_{2st}=43,56. Rezultatul este foarte apropiat de cel obținut prin liniarizarea sistemului de la punctul precedent.

Să determinăm acum răspunsul sistemului liniar (2.95) pompă – rezervor la un debit sinusoidal staționar de forma (2.77). Descompunem sinusoida în două componente complexe conform relației (2.83). Deoarece sistemul este liniar şi se aplică principiul superpoziției,

[69] În acest caz şi frecvența este zero iar perioada are o valoare infinită.

calculăm la început răspunsul la prima componentă (2.86). Răspunsul sistemului este tot sinusoida şi are forma (2.87). Înlocuind în (2.95) se obține:

$$Tj\omega_k Y_k e^{j\omega_k t} + Y_k e^{j\omega_k t} = KX_k e^{j\omega_k t} \tag{2.96}$$

Aceasta este o ecuație algebrică liniară şi putem calcula ieşirea Y_k în funcție de intrarea în sistem X_k.

$$Y_k = \frac{K}{1+Tj\omega_k} X_k = G_k X_k \tag{2.97}$$

în care G_k este raportul dintre ieşirea şi intrarea sistemului

$$G_k = \frac{Y_k}{X_k} = \frac{K}{1+Tj\omega_k} \tag{2.98}$$

G_k este numită ***constanta de transfer*** a sistemului liniar pentru intrarea sinusoidală cu pulsația ω_k. Această constantă este un număr complex a cărui valoare depinde de valoarea pulsației ω_k a semnalului sinusoidal care se propagă prin sistem.

Dacă se consideră în (2.98) toate valorile posibile ale pulsației $\omega=\omega_k$ se defineşte ***funcția de transfer în domeniul frecvență G(jω)*** a sistemului la propagarea semnalului sinusoidal de la intrarea sa la ieşirea lui în regim staționar:

$$G(j\omega) = \frac{K}{1+Tj\omega} = \left|G(j\omega)\right| e^{j\theta} \tag{2.99}$$

Modulul funcției de transfer este

$$\left|G(j\omega)\right| = \frac{K}{\sqrt{1+T^2\omega^2}} \tag{2.100}$$

Argumentul funcției de transfer este:

$$\theta = \arg G(j\omega) = -\arctan(\omega T) \tag{2.101}$$

Pentru a obține constanta de transfer a unui semnal sinusoidal cu pulsația ω_k se înlocuieşte această valoare în (2.99), (2.100)şi (2.101)

Folosind relațiile (2.84), (2.88), (2.97) şi (2.99) se poate determina valoarea semnalului X_k după propagarea lui prin sistem:

$$B_k = \left|G(j\omega_k)\right| A_k \tag{2.102}$$

$$\psi_k = \theta_k + \varphi_k \tag{2.103}$$

Datorită fidelității sinusoidale semnalul după propagarea prin sistem va fi de forma:

$$y(t) = B_k \cos(\omega_k t + \psi_k) \tag{2.104}$$

Se observă că ieşirea sinusoidală este complet determinată de (2.102) şi (2.103) dacă se cunoaşte modulul şi argumentul funcției de transfer pentru pulsația ω_k. Nu mai este necesar calculul răspunsului sistemului la a doua componentă X_{-k} a sinusoidei şi aplicarea, după aceia, a principiului superpoziției.

Să considerăm un exemplu numeric. Adoptăm aceleaşi valori ca şi în cazul precedent al unei intrări constante. Deci c=2, b=1, y_0=36. Din (2.94) şi (2.95) rezultă T=6 h şi K=1/6.

Funcția de transfer a sistemului este:

$$G(j\omega) = \frac{Y(j\omega)}{X(j\omega)} = \frac{6}{1+6j\omega} \tag{2.105}$$

Pentru ca modelul liniar să funcționeze este necesar ca semnalul de intrare să aibă valori mici în jurul punctului static de funcționare. Ca şi în cazul precedent alegem o sinusoida cu amplitudinea mică:

$$x(t) = 1{,}2\cos(\frac{1}{4}t - \frac{\pi}{2}) \tag{2.106}$$

Pulsația sinusoidei este ω_k=0,25. Înlocuind în (2.100), (2.101) şi folosind (2.102), (2.103) se obține valoarea sinusoidei la ieşirea din sistem

$$\left|G(j\omega_k)\right| = \frac{K}{\sqrt{1+T^2\omega_k^2}} = \frac{6}{\sqrt{1+6^2\cdot 0{,}25^2}} = 3{,}32 \tag{2.107}$$

$$\theta_k = \arctan(-\omega_k T) = -\arctan(0{,}25\cdot 6) = -0{,}98\ radiani \tag{2.108}$$

$$B_k = \left|G(j\omega_k)\right| A_k = 3{,}3\cdot 1{,}2 = 4 \tag{2.109}$$

Faza sinusoidei la ieşirea din sistem este

$$\psi_k = \theta_k + \varphi_k = -0{,}98 - \frac{\pi}{2} = -4{,}12 \text{ radiani} \tag{2.110}$$

Sinusoida după propagarea prin sistem va fi:

$$y(t) = 4\cos(\frac{1}{4}t - 4{,}12) \tag{2.111}$$

Pentru o sinusoidă cu pulsaţia ω_k=0 răspunsul este 6.1,2=7,2. Se observă că sistemul amplifică sinusoida cu pulsaţia ω_k=0,25 de aproape două ori mai puţin decât sinusoida cu pulsaţia zero şi deci acţionează ca un filtru trece jos. Întârzierea ieşirii sinusoidale faţă de intrarea sinusoidală este egală cu $\theta_k T/2\pi$=0,93 h.

2.12.5. Funcţia de transfer a filtrului analogic de ordin doi

In Fig. 2.29 se prezintă schematic un motor electric aşezat pe o fundaţie. Comportarea acestui sistem este echivalentă cu cea a unui filtru analogic trece jos de ordin doi. Acest filtru lasă să treacă prin sistem numai semnalele sinusoidale cu frecvenţa joase.

Echilibrul forţelor conduce la următorul model matematic pentru acest sistem:

$$x(t) = m\frac{d^2y(t)}{dt^2} + c\frac{dy(t)}{dt} + ky(t) \tag{2.112}$$

Variaţia în timp a forţei este sinusoidală de forma (2.77). Datorită faptului că sistemul este sinusoidal se bucură de proprietatea de fidelitate sinusoidală şi ieşirea sistemului, deplasarea fundaţiei este tot sinusoidală. Aplicând aceiaşi metodă ca şi în paragraful precedent rezultă următoarea funcţie de transfer în domeniul frecvenţă pentru acest sistem:

$$G(j\omega) = \frac{1}{m(j\omega)^2 + c(j\omega) + k} = K\frac{\omega_n^2}{(j\omega)^2 + 2\zeta\omega_n(j\omega) + \omega_n^2} \tag{2.113}$$

în care ω_n este pulsaţia naturală iar ζ este fracţiunea de amortizare critică.

Această funcție de transfer de ordinul doi este în acelaşi timp şi modelul matematic standard cu ajutorul căruia se definesc performanțele unui sistem de reglare automată.

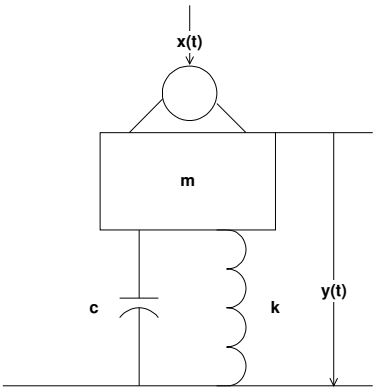


Fig. 2.29. Oscilatorul mecanic.

Programul de simulare pentru un sistem descris de un model matematic tip ecuație diferențială de ordinul doi cu coeficienți constanți este prezentat în *Fig. 2.30*. Graficul modulului şi fazei funcției de transfer în domeniul frecvență apare în *Fig. 2.31*.

```
//filtru2poli.sci
clear
s=poly(0,'s');                  //definirea variabilei s folosite in polinom
g=1/(1+0.44*s+s^2);             //definirea raport doua polinoame
G=syslin('c',g);                //definirea functiei de transfer
[f,repf]=repfreq( G,0.01,0.5); //calcul raspuns in frecventa, 100 frecvente
intre 0.01 si 0.1
xbasc();
[fi,db]=phasemag(repf,'m');    //calculul fazei si modulului
db=100*(10^(db/20));                    //trecerea din decibeli in amplitudine
amplificata de 10 ori
plot2d([f',f'],[fi',db']);          //tiparire doua grafice
xgrid();
xtitle('Analiza in domeniul frecventa - filtrul cu doi poli', 'f[Hz]',...
'100*Modul, Faza[grade]');
```

Fig. 2.30 Programul Scilab pentru calculul modulului şi fazei funcției de transfer

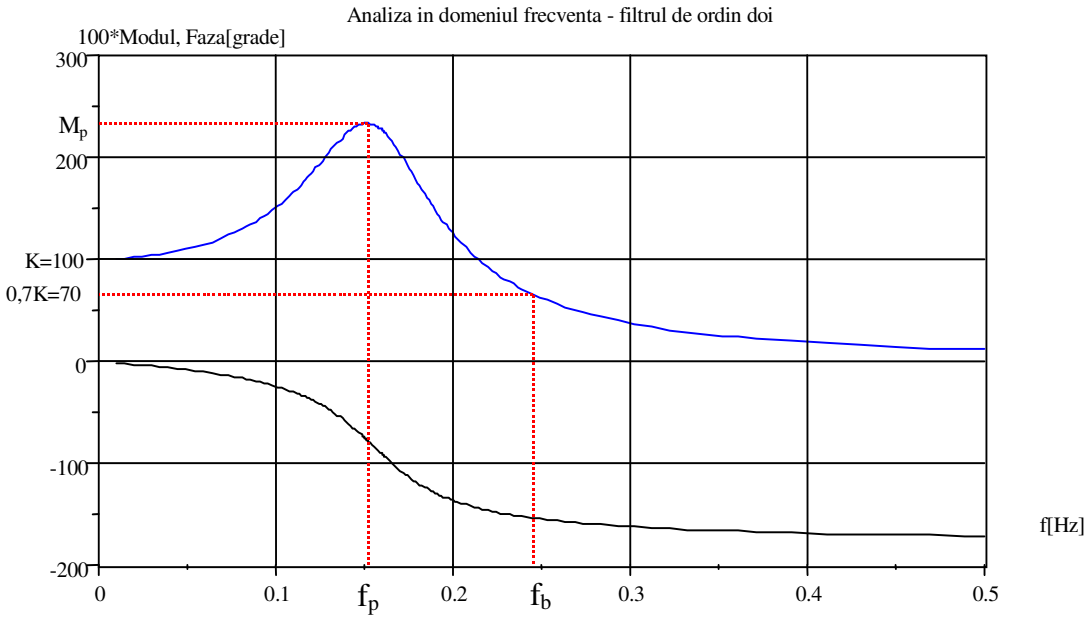


Fig. 2.31 Graficul modulului şi fazei funcţiei de transfer

În *Fig. 2.31* există câteva puncte de referinţă de referinţă. Conform relaţiei (2.113) pulsaţiei ω=0 îi corespunde valoarea K. Prin definiţie punctul corespunzător valorii 0,707K defineşte banda de trecere a filtrului caracterizată de frecvenţa f_p. Amplificarea maximă a filtrului este M_p şi apare la pulsaţia ω_p. Relaţiile dintre aceste mărimi şi fracţiunea de amortizare critică ζ şi pulsaţia naturală ω_n sunt prezentate în continuare.

$$M_p = \frac{1}{2\zeta\sqrt{(1-\zeta^2)}} \qquad (2.114)$$

$$\omega_p = \omega_n\sqrt{(1-2\zeta^2)} \qquad (2.115)$$

Din motive practice graficul din *Fig. 2.31* este prezentat în coordonate logaritmice. Pe ordonată se foloseşte unitatea de măsură decibelul. Modulul exprimat în decibeli se obţine cu ajutorul următoarei relaţii:

$$M_{dB} = 20\log_{10}(M) \qquad (2.116)$$

2.12.6. Semnale periodice. Seria Fourier

Semnalele periodice cu perioada T_0 pot fi exprimate ca o sumă de sinusoide cu ajutorul seriei Fourier.

$$x(t) = \frac{a_0}{2} + \sum_{k=1}^{k=\infty} (a_k \cos k\omega_0 t + b_k \sin k\omega_0 t) \qquad (2.117)$$

$$a_0 = \frac{2}{T_0} \int_{t_0}^{t_0+T_0} x(t)dt \qquad (2.118)$$

$$a_k = \frac{2}{T_0} \int_{t_0}^{t_0+T_0} x(t)\cos(k\omega_0 t)dt \qquad (2.119)$$

$$b_k = \frac{2}{T_0} \int_{t_0}^{t_0+T_0} x(t)\sin(k\omega_0 t)dt \qquad (2.120)$$

Folosind relația lui Euler se poate obține seria Fourier sub formă complexă:

$$x(t) = \sum_{k=-\infty}^{\infty} X_k e^{jk\omega_0 t} \qquad (2.121)$$

$$X_k = \frac{1}{T_0} \int_{t_0}^{t_0+T_0} x(t)e^{-jk\omega_0 t} \qquad (2.122)$$

Semnalul periodic $x_P(t)$ din *Fig. 2.31*a) a fost obținut prin repetarea periodică, cu perioada T_0, a unui impuls x(t) cu forma unui dinte de ferestrău. Modulul componentelor X_k este prezentat în *Fig. 2.32*b).

2.12.7. Semnale impuls. Transformarea Laplace. Funcția de transfer.

Dacă perioada T_0 de repetiție a dintelui de ferestrău crește spre infinit semnalul periodic se transformă într-un impuls iar liniile spectrului din *Fig. 2.32*b) se apropie formând

o funcție continuă. Suma din (2.121) se transformă în integrală şi se obține transformarea Fourier determinată prin relațiile următoare:

$$X(j\omega) = F\{x(t)\} = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \qquad (2.123)$$

$$x(t) = F^{-1}\{X(j\omega)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\omega)e^{j\omega t}d\omega \qquad (2.124)$$

În *Fig. 2.32* se observă că periodizarea în domeniul timp duce la o eşantionare în domeniul frecvență. Reciproca este de asemenea valabilă. O eşantionare în domeniul timp conduce la o periodizare în domeniul frecvență. Dacă spectrul continuu este infinit, periodizarea sa conduce la o suprapunere a valorilor provocând fenomenul de aliere sau suprapunere. Dacă din acest spectru cu suprapuneri se încearcă obținerea semnalului inițial se obțin rezultate eronate. Evitarea acestei erori se face impunând condiția ca frecvența de eşantionare să fie mai mare decât de două ori frecvența maximă din spectrul continuu.

$$f_e = 2f_{\max} \qquad (2.125)$$

Relația (2.125) este numită teorema eşantionării, teorema lui Shannon sau teorema lui Nyquist. Frecvența $f_e/2$ este numită adeseori frecvența Nyquist.

În *Fig. 2.32*c) frecvența de eşantionare este $f_e=1/T$. În domeniul frecvență din *Fig. 2.32*d) această frecvență, sau pulsația corespunzătoare Ω, măsoară perioada de repetiție a spectrului. Se observă că Ω este de două ori mai mare decât pulsația maximă a spectrului

Din relațiile (2.122) şi (2.123) rezultă:

$$X(j\omega) = \lim_{T_0\to\infty}\frac{X_k}{f} \qquad (2.126)$$

Datorită acestei legături între spectrul X_k şi transformarea Fourier, aceasta din urmă se mai numeşte şi *densitate spectrală*.

Integrala (2.123) nu este convergentă pentru unele semnale frecvent întâlnite în practică, de exemplu treapta unitară. Acest dificultate este depăşită dacă se înlocuieşte jω cu frecvența complexă s definită astfel:

$$s = \sigma + j\omega \qquad (2.127)$$

Relația (2.123) se transformă în transformata Laplace bilaterală:

$$X(s) = L\{x(t)\} = \int_{-\infty}^{\infty} x(t)e^{-st}\,dt \qquad (2.128)$$

Transformarea Laplace inversă va fi:

$$x(t) = L^{-1}\{X(s)\} = \frac{1}{2\pi j}\int_{s=\sigma_0-j\infty}^{\sigma_0+j\infty} X(s)e^{st}\,ds \qquad (2.129)$$

Multe semnale x(t) au însă un început, adică un moment înaintea căruia sunt nule. Considerând originea timpului în acel moment se obține transformata Laplace unilaterală:

$$X(s) = L\{x(t)\} = \int_{0}^{\infty} x(t)e^{-st}\,dt \qquad (2.130)$$

Transformarea Laplace inversă (2.129) rămâne aceiaşi şi pentru transformarea Laplace unilaterală. Diferența dintre ele apare atunci când sistemul prin care se propagă semnalul are condiții inițiale diferite de zero. Pentru condiții inițiale nule cele două transformări sunt identice.

Funcția de transfer a unui filtru pentru semnale impuls se obține din funcția de transfer la semnale sinusoidale prin înlocuirea lui jω cu s. De exemplu, funcția de transfer (2.113) a filtrului de ordinul doi devine:

$$G(s) = \frac{Y(s)}{X(s)} = K\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \qquad (2.131)$$

Funcția de transfer se defineşte, de obicei, sub forma cea mai generală prin raportul dintre transformata Laplace a semnalului de ieşire şi a transformatei Laplace a semnalului de intrare în condiții inițiale nule. Înlocuind pe s cu jω se obține funcția de transfer definită cu ajutorul transformatei Fourier prin raportul densităților spectrale ale semnalelor de ieşire şi intrare.

$$G(j\omega) = \frac{Y(j\omega)}{X(j\omega)} = K\frac{\omega_n^2}{(j\omega)^2 + 2\zeta\omega_n(j\omega) + \omega_n^2} \qquad (2.132)$$

Dacă semnalele de intrare şi ieşire sunt sinusoidale atunci funcția de transfer se obține din forma generală (2.131) înlocuind pe s cu $j\omega_k$ şi se obține:

$$G_k = \frac{Y_k}{X_k} = K \frac{\omega_n^2}{(j\omega_k)^2 + 2\zeta\omega_n(j\omega_k) + \omega_n^2} \qquad (2.133)$$

De data aceasta funcția de transfer este definită prin raportul sinusoidelor complexe (2.86) de intrare şi ieşire.

În practica prelucrării datelor transformatele Fourier şi Laplace sunt calculate prin reducerea lor la serii Fourier. Pentru aceasta semnalele impuls sunt periodizarea şi eşantionate ca în *Fig. 2.32*. Se foloseşte algoritmul transformării Fourier rapide.

Trecerea semnalelor prin sisteme liniare, de exemplu filtre, provoacă modificarea lor prin efectuarea asupra lor a unor operații. Efectul acestor operații în domeniul timp şi în domeniul frecvență este prezentat în *.Tab. 2.1* Se observă că transformarea Laplace unilaterală ia în considerare în mod *explicit* existența energiilor acumulate în elementele sistemului şi caracterizate prin existența unor condiții inițiale diferite de zero. Transformarea Laplace bilaterală, ca şi transformarea Fourier care constituie un caz particular al ei, trebuie să considere separat semnalele datorate energiilor acumulate inițial. Dacă sistemul este relaxat, adică nu are energie acumulată în elemente şi deci condițiile inițiale sunt nule, atunci transformările Laplace bilaterală şi unilaterală sunt echivalente.

Tab 2.1 Prelucrarea semnalelor în domeniul timp şi în domeniul frecvență

Operatia	*Domeniul timp x(t)*	*Domeniul frecventa*	
		Transformarea Laplace bilaterala X(s)=L{x(t)}	*Transformarea Laplace unilaterala X(s)=L{x(t)}*
Derivarea	$\frac{dx(t)}{dt}$	$sX(s)$	$sX(s) - x(0^-)$
Integrarea	$\int x(t)dt$	$\frac{1}{s}X(s)$	$\frac{1}{s}X(s) + \frac{1}{s}\left[\int x(t)dt\right]_{t=0^-}$
Insumarea	$k_1x_1(t) + k_2x_2(t)$	$k_1X_1(s) + k_2X_2(s)$	
Intarzierea	$x(t-\tau)$	$e^{-s\tau}X(s)$	
Convolutia	$\int_{-\infty}^{t} x(\tau)h(t-\tau)d\tau$	$H(s)X(s)$	
Schimbare scara	$X(\frac{t}{\lambda})$	$\lvert\lambda\rvert X(\lambda s)$	

În *Tab. 2.1* se observă că operațiile de derivare şi integrare a semnalelor se transformă în domeniul frecvență în operații de înmulțire şi împărțire care sunt mult mai simplu de realizat.

În *Tab. 2.2* sunt prezentate Transformatele Laplace ale semnalelor cel mai des întâlnite în practică. Se observă că acestea sunt înmulțite în domeniul timp cu funcția treaptă unitara asigurând prin aceasta începerea lor la momentul de timp egal cu zero.

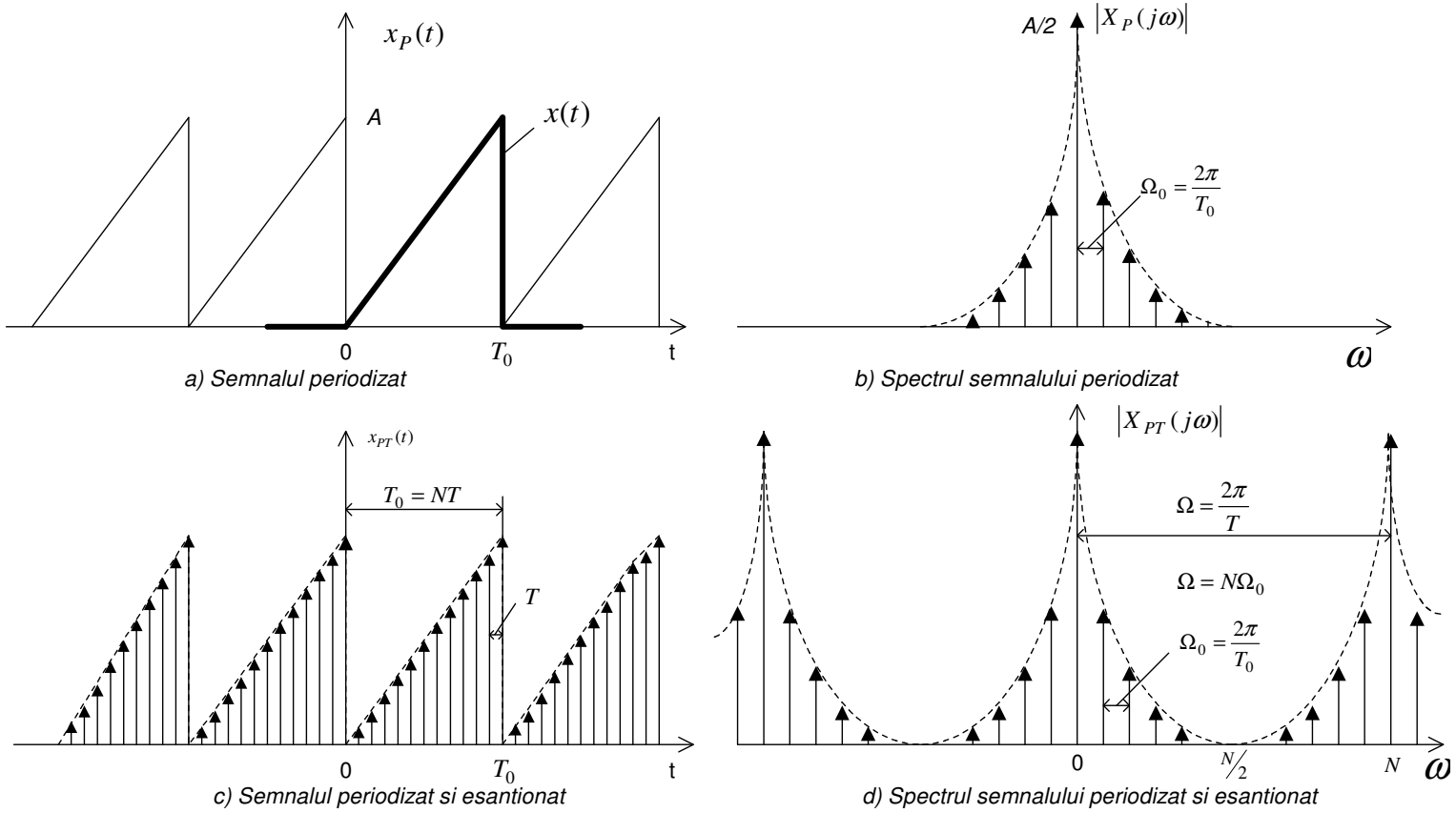
Tab 2.2 Transformate Laplace ale semnalelor

$x(t)$	$X(s)$
$\delta(t)$	1
$u^1(t)$	$\frac{1}{s}$
$u^1(t)t^n$	$\frac{n!}{s^{n+1}}$
$u^1(t)e^{-\alpha t}$	$\frac{1}{(s+\alpha)}$
$u^1(t)e^{-\alpha t}\sin\beta t$	$\frac{\beta}{(s+\alpha)^2+\beta^2}$
$u^1(t)e^{-\alpha t}\cos\beta t$	$\frac{s+\alpha}{(s+\alpha)^2+\beta^2}$
$u^1(t)\left\{1-e^{-\frac{t}{T}}(1+\frac{t}{1!T}+\frac{t^2}{2!T^2}+\cdots+\frac{t^{n-1}}{(n-1)!T^{n-1}})\right\}$	$\frac{1}{(1+Ts)^n}\frac{1}{s}$
$u^1(t)\left\{1-\frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\cos\left[(\omega_n\sqrt{1-\zeta^2})t+arctg\frac{\zeta}{\sqrt{1-\zeta^2}}\right]\right\}$	$\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}\frac{1}{s}$

În *Tab. 2.3* sunt prezentate transformatele Fourier ale unor semnale frecvent întâlnite în practică. Pentru semnalele impuls care încep de la zero transformata Fourier este identică cu Transformata Laplace din *Tab. 2.2* dacă se înlocuieşte s cu jω.

Tab 2.3 Transformatele Fourier ale semnalelor

$x(t)$	$X(j\omega)$
$\delta(t)$	1
$u^1(t)$	$\pi\delta(\omega)+\frac{1}{j\omega}$
$u^1(t)e^{-\alpha t}$	$\frac{1}{j\omega+\alpha}$
$u^1(t)e^{-\alpha t}\sin\beta t$	$\frac{\beta}{(j\omega+\alpha)^2+\beta^2}$
$u^1(t)e^{-\alpha t}\cos\beta t$	$\frac{j\omega+\alpha}{(j\omega+\alpha)^2+\beta^2}$
$e^{j\alpha t}$	$2\pi\delta(\omega-\alpha)$
1	$2\pi\delta(\omega)$
$\cos\beta t$	$\pi[\delta(\omega-\beta)+\delta(\omega+\beta)]$
$\sin\beta t$	$\frac{\pi}{j}[\delta(\omega-\beta)-\delta(\omega+\beta)]$



a) Semnalul periodizat *b) Spectrul semnalului periodizat*

c) Semnalul periodizat si esantionat *d) Spectrul semnalului periodizat si esantionat*

Fig. 2.32 Efectul periodizării şi eşantionării semnalelor

2.12.8. Semnificații fizice în domeniul frecvență

Studiul în domeniul frecvență al semnalelor este folosit frecvent în practică datorită avantajelor sale dar are un caracter mai puțin intuitiv. Din această cauză legăturile directe între domeniul timp şi domeniul frecvență prezintă o importanță deosebită

Variabilele independente în cele două domenii sunt timpul şi frecvența. În domeniul frecvență se foloseşte frecvent drept variabilă independentă şi pulsația ω.

$$\omega = 2\pi f = \frac{2\pi}{T}\left[rad/s\right] \qquad (2.134)$$

în care f este frecvența [Hz] iar T [s] este perioada sinusoidelor de intrare şi ieşire.

Transformatele Laplace sau Fourier ale semnalelor au dimensiunea semnalului original înmulțită cu timpul datorită relațiilor de definiție.

Între domeniul timp şi domeniul frecvență există următoarele relații directe.

Teorema valorii finale.

$$x(\infty) = \lim_{t\to\infty} x(t) = \lim_{s\to 0}\left[sX(s)\right] \qquad (2.135)$$

Teorema valorii inițiale.

$$x(0) = \lim_{t\to 0} x(t) = \lim_{s\to\infty}\left[sX(s)\right] \qquad (2.136)$$

Teorema lui Parceval.

$$\int_{-\infty}^{\infty} x^2(t)dt = \frac{1}{2\pi}\int_{-\infty}^{\infty}\left|X(j\omega)\right|^2 d\omega \qquad (2.137)$$

în care $\left|X(j\omega)\right|$ este modulul transformării Fourier.

Integrala din stânga a relației este proporțională cu energia semnalului. Din această cauză $\left|X(j\omega)\right|^2$, o mărime măsurabilă uşor, se numeşte densitate spectrală a energiei.

2.12.9. Răspunsul sistemelor la semnale impuls în domeniul frecvență

Pentru un sistem liniar, datorită aplicabilității principiului superpoziției, dacă se cunoaşte răspunsul forțat al sistemului la o intrare particulară atunci se poate calcula răspunsul forțat al sistemului la orice intrare. De exemplu dacă se cunoaşte funcția pondere h(t), care este răspunsul forțat al sistemului la un impuls Dirac, integrala de convoluție permite calculul răspunsului pentru o intrare oarecare.

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau = h(t) * x(t) \qquad (2.138)$$

Aplicând transformata Laplace acestei relații se obține conform Tab. 3.1:

$$Y(s) = L\{h(t) * x(t)\} = H(s)U(s) \tag{2.139}$$

în care H(s) este transformata Laplace a funcției pondere şi coincide cu funcția de transfer definită în paragrafele precedente folosind semnale sinusoidale. Relații asemănătoare se pot definii şi pentru alte semnale tip de intrare, de exemplu funcția treaptă unitară.

Funcția de transfer se determină experimental sau analitic aplicând transformata Laplace ecuațiilor diferențiale care reprezintă modelul sistemului. Să prezentăm un exemplu.

Modelul (2.95) al sistemului pompă – rezervor poate avea următoarea realizare:

$$5\frac{dy(t)}{dt} + y(t) = x(t) \tag{2.140}$$

cu nivelul inițial în rezervor

$$y(0) = 0{,}14 \tag{2.141 }$$

Aplicând transformata Laplace ecuației (2.140) rezultă:

$$sY(s) - y(0) = -0{,}2Y(s) + 0{,}2X(s) \tag{2.142}$$

Răspunsul la o intrare oarecare X(s) va fi:

$$Y(s) = \frac{0{,}2}{s + 0{,}2}X(s) + \frac{y(0)}{s + 0{,}2} \tag{2.143}$$

Pentru condiții inițiale nule rezultă funcția de transfer:

$$H(s) = \frac{0{,}2}{s + 0{,}2} = \frac{1}{5s + 1} \tag{2.144}$$

După determinarea funcției de transfer pe cale experimentală sau analitică, cum a fost cazul aici, de determină din tabele, de exemplu *Tab. 2.2*, Transformarea Laplace X(s) pentru intrarea dorită x(t). Dacă ,se doreşte răspunsul la o treaptă unitară $x(t)=u^1(t)$ se obține:

$$X(s) = \frac{1}{s} \tag{2.145}$$

Introducând această valoare şi condiția inițială (2.139) în (2.143) rezultă răspunsul sistemului la o treaptă de unitară de intrare care poartă denumirea de răspuns indicial.

$$Y(s) = \frac{0,2}{s+0,2}\frac{1}{s} + \frac{0,14}{s+0,2} = Y_f(s) + Y_n(s) \qquad (2.146)$$

în care $Y_f(s)$ este răspunsul forțat iar $Y_n(s)$ este răspunsul natural al sistemului. Prin aplicarea principiului superpoziției cele două componente ale răspunsului pot fi prelucrate separat. Trecerea din domeniul frecvență în domeniul timp se face tot cu ajutorul tabelelor. În *Tab. 2.2* componentei naturale îi corespunde:

$$y_n(t) = 0,14u^1(t)e^{-0,2t} \qquad (2.147)$$

Componenta forțată $Y_f(s)$ nu are însă o corespondență directă în*Tab. 2.2*. Există însă tabele de transformate Laplace mai complete. O altă cale de rezolvare a problemei constă în folosirea *Tab. 2.2* şi descompunând pe $Y_f(s)$ în fracții simple.

$$Y_f(s) = \frac{0,2}{s+0,2}\frac{1}{s} = \frac{A}{s} + \frac{B}{s+0,2} \qquad (2.148)$$

Aducând la acelaşi numitor şi identificând coeficienții variabilei s din (2.149) rezultă A=1 şi B=-1. Cu $Y_f(s)$ descompus se poate folosi *Tab. 2.2* şi rezultă:

$$y_f(t) = u^1(t) - u^1(t)e^{-0,2t} = u^1(t)[1 - e^{-0,2t}] \qquad (2.149)$$

Suprapunând răspunsul forțat şi răspunsul natural se obține:

$$y(t) = y_f(t) + y_n(t) = u^1(t)[1 - e^{-0,2t}] + 0,14u^1(t)e^{-0,2t} = u^1(t)[1 - 0,86e^{-0,2t}] \qquad (2.150)$$

2.13. Noțiuni de proiectre a sistemelor automate

Proiectrea sistemelor automate constă în elaborarea documentației pentru execuția sistemelor automate care sunt definite de:

- Elementele componente. Elementele tipice ale unui sistem automat sunt elementele de conducere (automatele cu contacte şi relee, automatele programabile, regulatoare, controlere, calculatoare de proces), elementele de execuție (robinete de reglare, pompe, contactoare), elementele de măsură (traductoare, detectoare).
- Structura sistemului automat. Câteva structuri tipice sunt: structura secvențială, structura cu paralelism, structura secvențială cu paralelism tip bandă rulantă, bucla simplă, bucla cu două grade de libertate, bucla cu feedforward.
- Scopul sistemului automat. Cele mai frecvente scopuri pentru care sunt definite sistemele automate sunt: înlăturarea perturbațiilor, urmărirea referinței, compensarea regimului tranzitoriu, micşorarea sensibilității la defecțiuni, liniarizarea, optimizarea unor criterii.

Proiectarea sistemelor automate are două etape principale succesive: proiectarea funcțională şi proiectarea constructivă.

2.13.1. Proiectarea funcțională

În cadrul proiectării funcționale se defineşte sistemul automat caracterizat de elemente, structură şi scop.

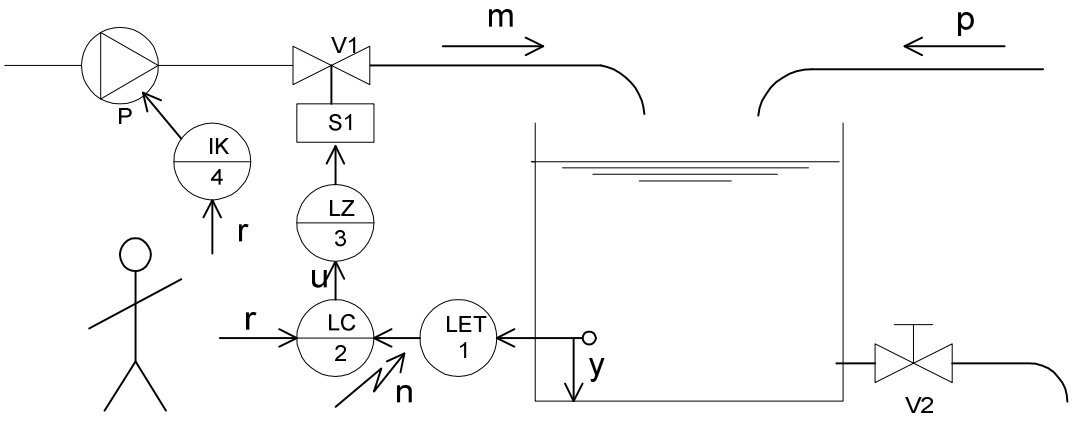


Fig. 2.33 O schemă tehnologică cu echipamentele de automatizare formate dintr-un automat şi o buclă de reglare a nivekului.

Rezultatul proiectării funcționale îl constitue: 1) schema tehnologică cu echipamentele de automatizare, de exemplu cea din Fig. 2.33, 2) parametrii elementelor componente,

inclusiv algoritmul de conducere şi 3) performanţele sistemului automat. Pentru obţinerea acestui rezultat se fac experimentări, calcule şi simulări.

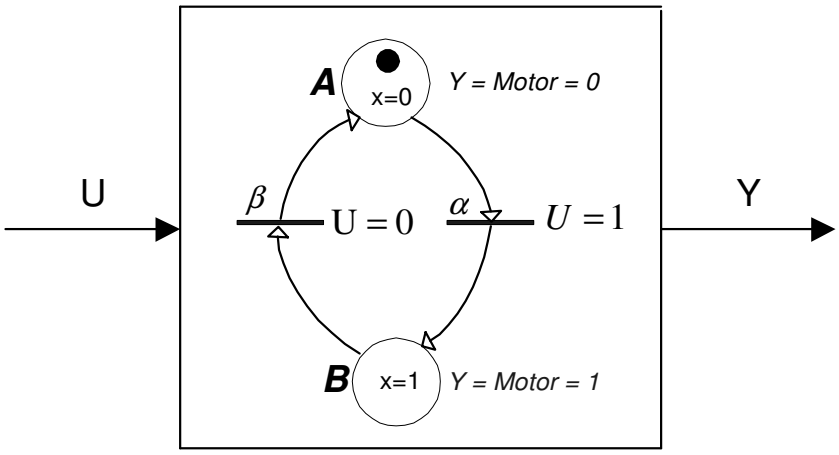


Fig. 2.34 Modelul unui motor sub formă de reţea Petri

În cadrul proiectării funcţionale a sistemului automat un loc central îl ocupă proiectarea echipamentelor de conducere. Pentru realizarea aceasta este necesară cunoaşterea unor modele a echipamentelor şi proceselor conduse.

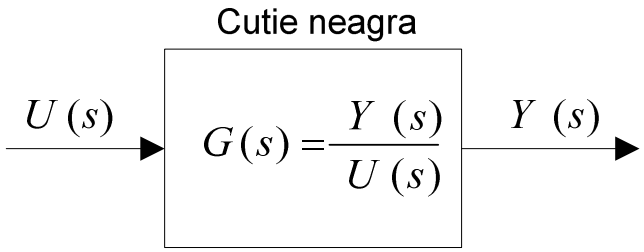


Fig. 2.35 Modelul unui proces continuu din pompă.

Pentru pompa P din Fig. 2.33 se pot elabora mai multe modele, în funcţie de scopul conducerii şi performanţele dorite. Dacă se doreşte doar automatizarea pornirii şi opririi motorului pompei se poate elabora un model cu evenimente discrete de tip reţea Petri ca în Fig. 2.34. Pompa împreună cu motorul care o antrenează formează un echipament complex în care au loc diferite procese de natură hidraulică, mecanică şi electrică. În cazul în care scopul conducerii automate îl constituie reglarea continuă a turaţiei Y(s) a pompei în funcţie de tensiunea de alimentare U(s) atunci se elaborează un model sub formă de funcţie de transfer pentru un proces ca în Fig. 2.35.

În funcţie de modelul ales se determină modelul elementului de conducere. Dacă modelul este o reţea Petri atunci elementul de conducere va fi un automat care va genera mărimea de comandă U din Fig. 2.34 astfel încât să se obţină valorile dorite pentru Y. Reţeaua

Petri pentru acest automat este prezentată în Fig. 2.36. Intrările sunt două butoane în impuls p şi o pentru pornire şi oprire iar ieşirea este constituită din tensiunea de comandă u.

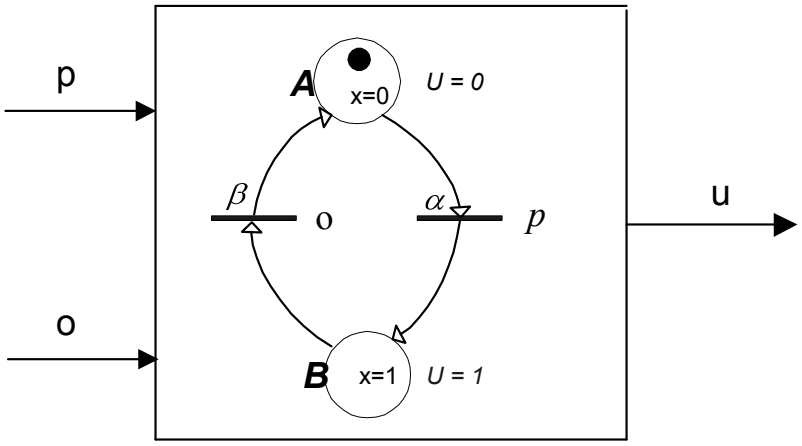


Fig. 2.36 Modelul sub formă de reţea Petri a automatului pentru motorul din Fig. 2.34.

Pentru conducerea procesului din Fig. 2.35 se foloseşte un compensator K prezentat în Fig. 2.37. Dacă funcția de transfer K a compensatorului este egală cu inversa funcției de transfer a procesului atunci ieşirea Y este egală cu referința R deoarece în cazul acesta funcția de transfer a sistemului este egală cu produsul funcțiilor de transfer a elementelor componente.

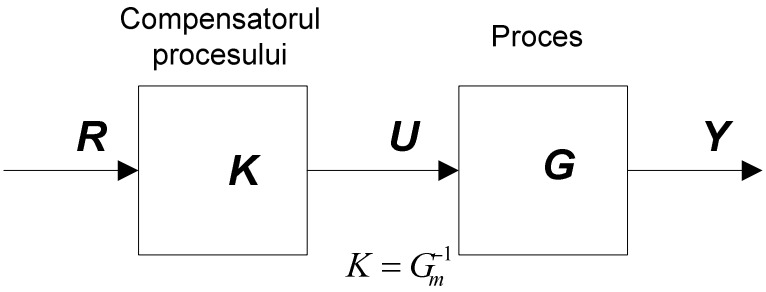


Fig. 2.37 Conducerea unui proces cu ajutorul compensatorului

Valoare obținută pentru funcția de transfer a compensatorului este adeseori nerealizabilă fizic şi din această cauză se recurge la soluția din Fig. 2.38 în care funcția de transfer G_m este un model al funcției de transfer a procesului G. Dacă compensatorul erorii are funcția de transfer tip constantă de proporționalitate cu o valoare foarte mare $K_c=\infty$, atunci cele două soluții din Fig. 2.37 şi Fig. 2.38 sunt echivalente. Se observă că dacă $G_m=G$ atunci schema din Fig. 2.38 devine echivalentă cu cea din Fig. 2.39 care reprezintă bucla de reglare standard. Folosind structura din Fig. 2.39 se obțin si alte performanțe în afară de faptul că ieşirea este egală cu referință. Bucla de reglare poate să amelioreze puternic efectele

perturbațiilor, al zgomotelor de măsură şi al variațiilor mici pentru funcția de transfer G a procesului.

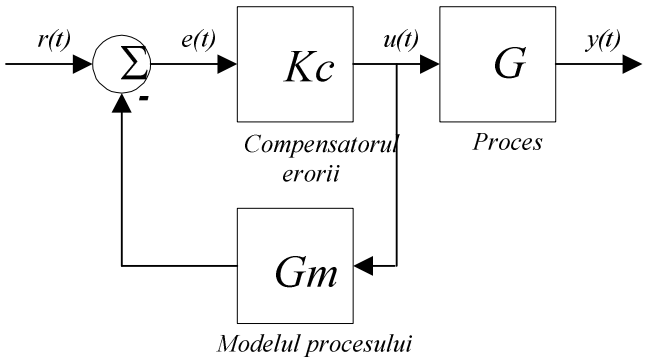


Fig. 2.38 Realizarea compensatorului procesului cu ajutorul unei bucle.

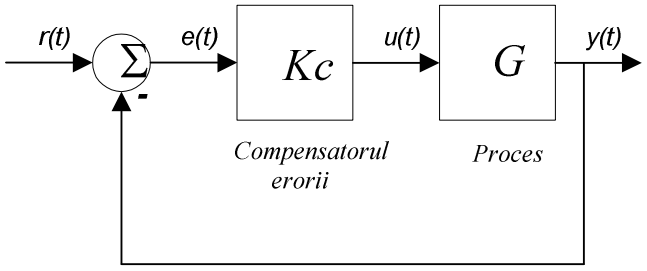


Fig. 2.39 Bucla de reglare standard

2.13.2. Proiectarea constructivă

Sistemul automat definit în etapa de proiectare funcțională trebuie implementat (materializat). Există diferite variante de implementare. De exemplu, automatul poate fi implementat cu contacte şi relee sau cu anjutorul unui automat programabil logic. Regulatorul poate fi implementat cu ajutorul unui regulator direct, un regulator electronic sau un calculator de proces. Documentația pentr această implementare se realizează în etapa de proiectare constructivă. De exemplu, programul automatului pentru motor se obține din rețeaua Petri din Fig. 2.36 sub forma unui grafcet prezentat în Fig. 2.40. Acest grafcet se introduce direct, sub formă grafică, în automatul programabil logic. Dacă automatul se implementează cu contacte şi relee atunci în etapa de proiectare constructivă se realizează schema electrică desfăşurată din Fig. 2.41.

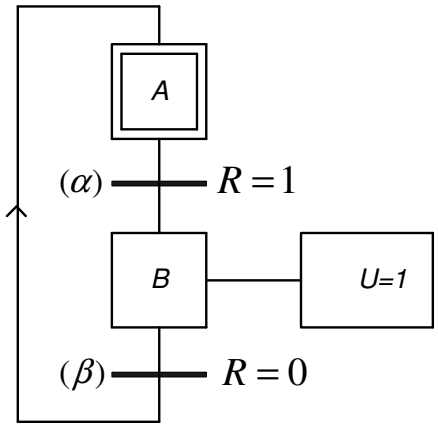


Fig. 2.40 Programul automatului programabil logic pentru conducerea motorului caracterizat de rețeaua Petri din Fig. 2.34.

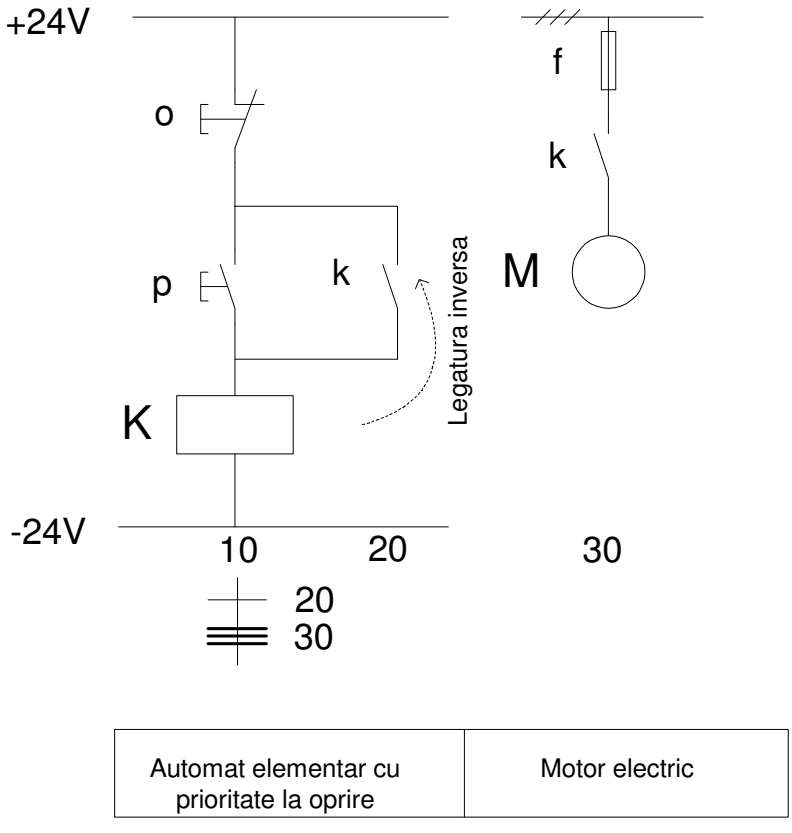


Fig. 2.41 Schema desfăşurată electrică pentru automatul motorului realizat cu contacte şi relee.

La încheierea acestui capitol introductiv reamintim câteva concepte fundamentale.

Sistemul – sau instalaţia, într-o accepţiune inginerească mai restrânsă – este cu conducere automată[70] dacă are *memorie* şi / sau cel puţin o *legătură inversă (reacţie, feedback) negativă*[71]. Algoritmii de conducere cei mai frecvenţi sunt cei de tip automat secvenţial[72], P, PI, PD şi PID. Parametrii unui regulator PID sunt constanta de proporţionalitate K_R, constanta de timp integral T_I, constanta de timp derivativ T_D şi pentru regulatoarele numerice perioada de eşantionare T_e.

Un proiect de construire a unui sistem automat are o parte funcţională şi o parte constructivă. Proiectul va conţine *scheme tehnologice cu aparatură de automatizare, scheme bloc şi scheme desfăşurate electrice.* Obiectul automatizării îl constitue procesul şi nu instalaţia[73]. Într-o instalaţie automatizată pot exista mai multe bucle, fiecare destinată unui anumit proces.

Algoritmul de conducere pentru sistemele automate poate fi realizat cu ajutorul următoarelor aparate:

- automate secvenţiale cu contacte şi relee sau circuite electronice,
- automate programabile logice, APL,
- regulatoare directe de tip termostat, presostat sau nivostat,
- regulatoare electronice, pneumatice sau hidraulice,
- calculatoare (microcalculatoare) specializate numite controlere, automate programabile AP sau calculatoare de proces.

Unul dintre aceste aparate împreună cu elementul de execuţie şi traductorul poate forma o *buclă.* Proiectarea funcţională a sistemului automat format dintr-o bucă de reglare se face în domeniul timp sau în domeniul frecvenţă pe baza funcţiei de transfer definită drept raportul dintre transformata Laplace a ieşirii şi transformata Laplace a intrării cu condiţii iniţiale nule.

Cea mai importantă performanţă a sistemului automat o constitue stabilitatea. Dacă sistemul este suficent de stabil (robust) atunci se iau în considerare şi alte performanţe, dintre care foarte importantă este eroarea indicială în regim staţionar. Pentru multe procese regulatorul PI poate asigura eroarea indicială staţionară nulă.

---

[70] Pe scurt *sistem automat* sau *instalaţie automată.*

[71] Buclă.

[72] În exemplul prezentat în acest capitol automatul secvenţial a fost de tip bipoziţional – 2P.

[73] In exemplul din acest capitol instalaţia are două procese: un proces hidraulic cu mărimea de ieşire nivelul din rezervor şi un proces termic cu mărimea de ieşire temperatura apei din rezervor.

3. Sisteme logice combinaționale - SLC

Funcționarea multor dispozitive şi echipamente tehnice poate fi descrisă aproximativ cu ajutorul unor variabile binare, numite *biți,* care pot lua numai două valori: {acționat, neacționat}, {0, 1}, {oprit, pornit}, {stinsă, aprinsă}, etc. Cu aceste variabile se pot realiza diferite operații asemănătoare cu cele din logica propozițională, algebra booleană binară sau algebra părților unei mulțimi, după cum se vede în Tab. 3.1. În lucrările consacrate sistemelor de conducere automată se foloseşte o terminologie şi o notație împrumutată din aceste algebre şi din logică. În plus se adaugă unii termeni folosiți în tehnica circuitelor logice electronice integrate şi tehnica calculatoarelor electronice. Sistemele logice combinaționale nu sunt sisteme automate deoarece nu au feedback sau memorare. Ele stau însă la baza sistemelor automate la care elementul de conducere este de tip automat.

Tab. 3.1 Comparație între diferite algebre

Algebra părților unei mulțimi	***Algebra booleana binară***	***Logica propozițională***
Partea ***A*** a mulțimii ***E***	x ϵ {0, 1}	Enunț {fals, adevarat}
Intersecție (∩)	Înmulțire booleana (.)	Conector de coordonare (SI) notat (Λ)
Reuniune (U)	Adunare booleana (+)	Conector de coordonare (SAU) notat (V)
Complementare (¯)	Complementare (¯)	Conector de modificare (NU) sau negare notat (┐)

Modelul funcțional al unui sistem logic combinațional simplu este prezentat în Fig. 3.1. Se disting intrările *a* şi *b,* ieşirea *y* şi relația logică dintre ele *f.*

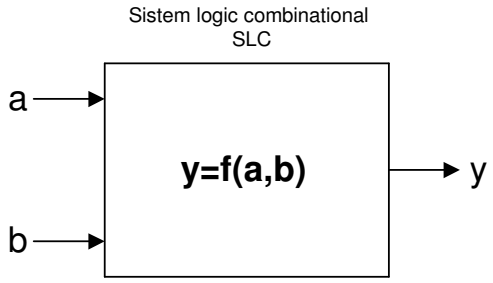


Fig. 3.1 Schema bloc a unui sistem logic combinațional

Se disting următoarele probleme care trebuiesc rezolvate în legătură cu sistemele logice combinaționale.

✓ ***Analiza***.

Se cunosc intrările *a, b* şi relația logică *f* şi se doreşte determinarea ieşirii *y*.

$$a, b, f \Rightarrow y \tag{3.1}$$

Pentru cunoaşterea relației logice *f* se aplică metoda *analitică*[74] care constă în descompunerea SLC în subsisteme cât mai simple şi stabilirea legăturilor dintre aceste subsisteme. În cazul SLC implementate cu contacte şi relee aceste subsisteme sunt de tipul *ŞI, SAU, NU*. Dacă SLC este realizat cu ajutorul circuitelor electronice integrate în afară de subsistemele menționate mai sunt şi altele, de exemplu circuitele *ŞI-NU, SAU-NU, SAU-EXCLUSIV, etc.*

Dacă se cunosc elementele componente ale SLC şi structura[75] sa, rezultă imediat relația logică *f*.

Există mai multe metode pentru rezolvarea problemei analizei. În primul rând se poate calcula ieşirea direct folosind metodele algebrei booleene. Altă metodă constă în determinarea ieşirilor componentelor SLC, combinarea lor conform structurii şi urmărirea în continuare a modului cum se propagă semnalele de la intrarea SLC la ieşirea sa.

✓ ***Sinteza.***

Se cunosc intrările *a, b* şi ieşirea *y* şi se determină relația logică *f* care la rândul ei permite determinarea sistemului logic combinațional. Sinteza este una dintre metodele inginereşti de proiectare pe lângă metoda exemplelor tip, metoda încearcă şi verifică şi altele.

$$a, b, y \Rightarrow f \tag{3.2}$$

✓ ***Testarea.***

Se cunosc relația logică *f* şi ieşirea dorită *y* şi se determină intrările necesare *a* şi *b*.

$$f, y \Rightarrow a, b \tag{3.3}$$

Această problemă trebuie rezolvată atunci când dorim să ştim dacă sistemul este defect, unde se găseşte defecțiunea şi care este tipul ei.

[74] Aplicarea acestei metode se numeşte analiză

[75] Relațiile dintre elemente.

3.1. Coduri

Unele echipamente şi instalaţii au o funcţionare care poate fi descrisă printr-un număr finit de stări. De exemplu, un motor reversibil poate fi oprit, se poate roti în sens direct sau în sens direct, un ascensor se poate afla lu unul dintre cele 12 etaje ale clădirii. Pentru acestea se foloseşte o codificare binară cu ajutorul mai multor biţi. Dacă avem n biţi putem codifica N stări cu ajutorul codului binar natural

$$N = 2^n \qquad (3.4)$$

De exemplu cele trei stări ale motorului reversibil pot fi codificate folosind un cod binar natura format din doi biţi *k1* şi *k2* care poate caracteriza, conform formulei (3.4), maximum patru stări. O codificare posibilă, nu singura, este prezentată în

Tab. 3.2. Codul binar natural cu patru biţi este dat în Tab. 3.3 iar codul binar distributiv cu patru biţi în Tab. 3.4. Se observă cd codul distributiv este mai intuitiv dar are posibilităţi de codificare mult mai mici.

*Tab. 3.2 Codificarea cu doi biţi **a** şi **b** a stărilor unui motor reversibil*

Starea motorului	***a***	***b***
oprit	0	0
rotire directă	0	1
rotire inversă	1	0
-	1	1

Tab. 3.3 Codul binar natural cu patru biți

a	***b***	***c***	***d***
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Tab. 3.4 Codul binar distributiv sau "1 din n"

a	***b***	***c***	***d***
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0

3.2. Analiza sistemelor logice combinaționale

Sistemele logice combinaționale ***SLC*** sunt formate dintr-o mulțime de elemente, relații şi scopuri. Sistemele se numesc combinaționale deoarece în funcționarea lor nu intervine variabila timp. Contează numai combinația (structura) elementelor. Fără considerarea timpului aceste sisteme nu pot avea memorie şi deci nu pot realiza operații automate, fără intervenția omului Elementele SLC pot fi caracterizate prin variabile binare care pot avea două valori {0,1}. Relațiile dintre elemente sunt de tip *ŞI, SAU, NU* iar scopul acestor sisteme îl constitue prelucrarea informației. Sistemele logice combinaționale au drept model teoretic o algebră booleană şi în continuare ne vom folosi de rezultatele obținute de acest model şi de terminologia specifică, Tab. 3.1. Elementele *SLC* sunt de natură electromecanică, electronică, hidraulică, etc. În continuare vom studia *SLC* formate din *contacte* şi *relee.* Variabilele binare ataşate contactelor vor fi notate cu litere mici iar cele care caracterizează funcționarea bobinelor, lămpilor, motoarelor, etc sunt notate cu litere mari.

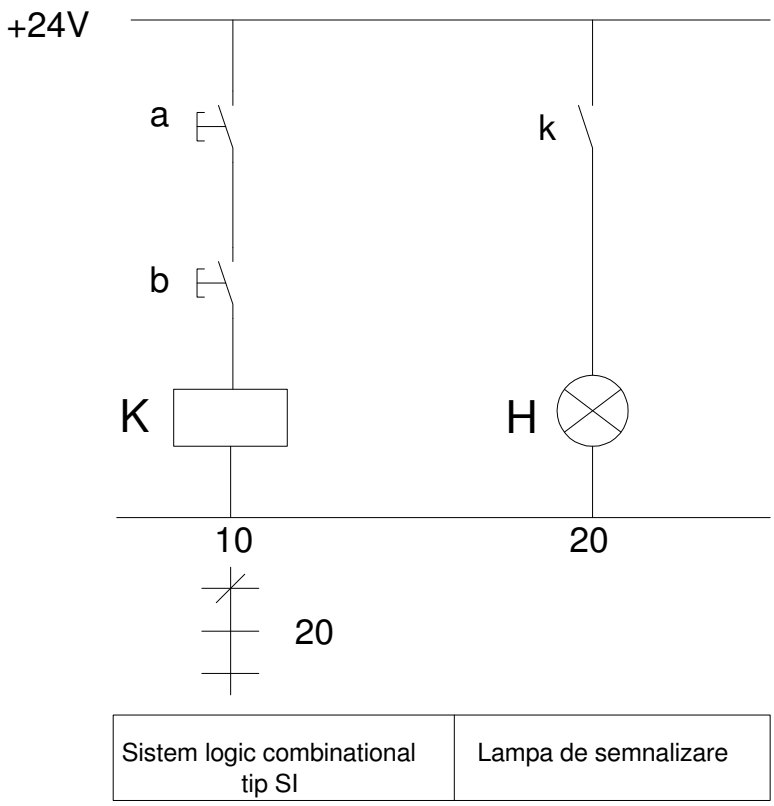


Fig. 3.2 Schema desfăşurată electrică a unui SLC de semnalizare tip ŞI

3.2.1. Sistem logic combinațional tip ŞI

În Fig. 3.2 se prezintă un sistem logic combinațional care are drept scop prelucrarea informației pentru semnalizare. Lampa *H* se aprinde numai dacă butoanele *a ŞI* b sunt acționate. Butonul este un contact normal deschis acționat manual. Dacă este neacționat prin el nu trece curent electric şi variabila care îl caracterizează are valoarea *0*. Atunci când este acționat prin el trece curent şi variabila are valoarea *1*. Bobina releului *K* are şi ea două valori: *1* atunci când prin ea trece curent şi *0* atunci când prin ea nu trece curent electric. Valoarea ei este o funcție de tip *ŞI* de valorile contactelor şi poate fi determinată *analizând* schema din Fig. 3.2. Prin *analiză* înțelegem că se cunoaşte SLC şi valorile semnalelor de intrare *a* şi *b* şi se determină teoretic, experimental sau prin simulare valorile semnalului de ieşire *K*. Pentru SLC cu contacte şi relee este foarte uşor să simulăm în imaginație funcționarea sistemului pentru toate valorile posibile ale semnalelor de intrare şi să trecem valorile obținute pentru semnalul de ieşire într-un tabel de adevăr cum este Tab. 3.5. Valorile contactelor *a* şi *b* pot fi trecute în orice ordine, important este să fie toate valorile posibile. Pentru a nu omite vre-o valoare vom folosi codul binar natural cu doi biți. Acesta are 2^n valori, în care n este numărul de biți. În cazul de față avem doi biți *a* şi *b* şi numărul de valori ale codului este patru. În primul rând al Tab. 3.5 contactele au valoarea *0* adică sunt neacționate. Prin ele nu poate să treacă curent electric şi deci şi curentul care trece prin bobina releului este nul. Deci rezultatul simulării indică valoarea *0* pentru variabila *K* care este trecută în ultima coloană. Se procedează asemănător pentru toate situațiile în care unul sau altul dintre butoane sunt acționate.

Tab. 3.5 Tabel de adevăr cu rezultatele analizei SLC tip ŞI

a	***b***	***K***
0	0	**?….0**
0	1	**?….0**
1	0	**?….0**
1	1	**?….1**

Releul electromagnetic are, într-o primă aproximare în care nu intervine timpul, următoarea relație dintre mărimea sa de intrare, starea *K* a bobinei şi mărimea sa de ieşire, contactul normal deschis *k*:

$$k = K \qquad (3.5)$$

Această relație spune că dacă trece curentul prin bobină şi deci releul este acționat atunci contactul său normal deschis este tot acționat. Considerând Tab. 3.5 şi (3.5) se poare definii relația logică de tip *ŞI* între contactele de intrare *a* şi *b* ale SLC şi contactul său de ieşire *k*.

$$k = a \cdot b \tag{3.6}$$

Conform Tab. 3.1 această operație mai este numită şi înmulțire booleană iar operatorul punct nu este specificat adeseori.

Funcția logică *ŞI* corespunde structurii de conectare în serie a contactelor.

Între lampa de semnalizare din Fig. 3.2 caracterizată prin variabila *H* şi contactul releului *k* există relația *H=k* adică lampa *H* este aprinsă atunci când contactul *k* este acționat şi considerând (3.6) se stabileşte relația logică a SLC de semnalizare prin acționarea manuală a butoanelor: *H=a.b.*

3.2.2. Sistem logic combinațional tip SAU

Contactele legate în paralel formează un sistem logic combinațional de tip SAU. Un exemplu cu două contacte este prezentat în Fig. 3.3. Analizând acest sistem prin simulare în imaginație într-un mod asemănător ca şi în cazul SLC tip ŞI se obține tabelul de adevăr Tab. 3.1. Lampa *H* este acționată (aprinsă) numai dacă sunt acționate butoanele *a SAU b.*

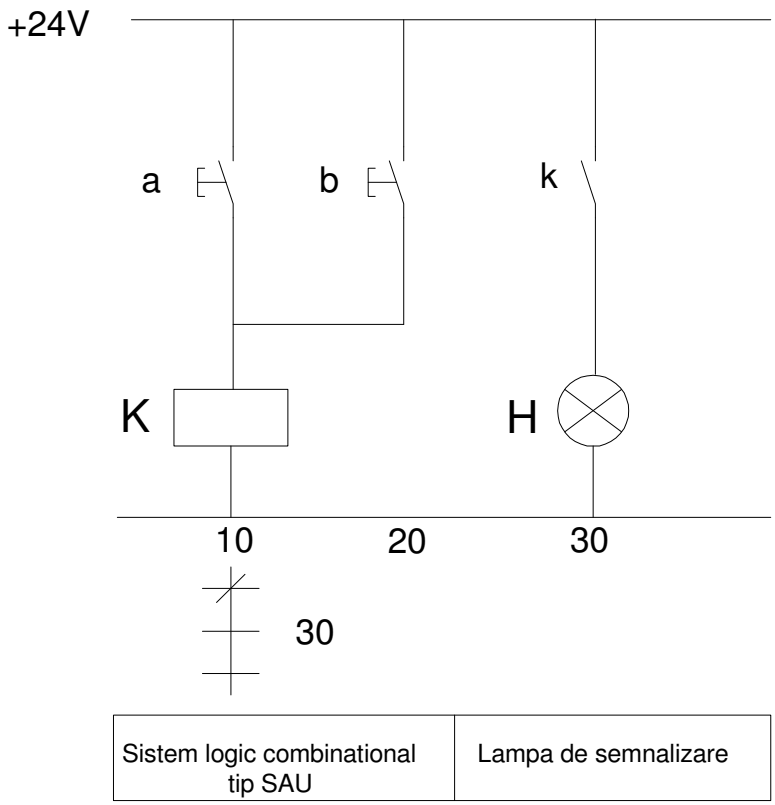


Fig. 3.3 Schema desfăşurată electrică pentru un SLC de semnalizare tip SAU

Tab. 3.6 Tabelul de adevăr cu rezultatele SLC tip SAU

a	***b***	***K***
0	0	**?….0**
0	1	**?….1**
1	0	**?….1**
1	1	**?….1**

Relația logică între contactele de intrare *a* şi *b* ale sistemului şi ieşirea sa *k* tip contact este:

$$k = a + b \tag{3.7}$$

3.2.3. Sistem logic combinațional tip NU

Sistemul logic combinațional tip NU are un singur contact ca în Fig. 3.4. Rezultatul analizei prin simulare este prezentat în Tab. 3.7.

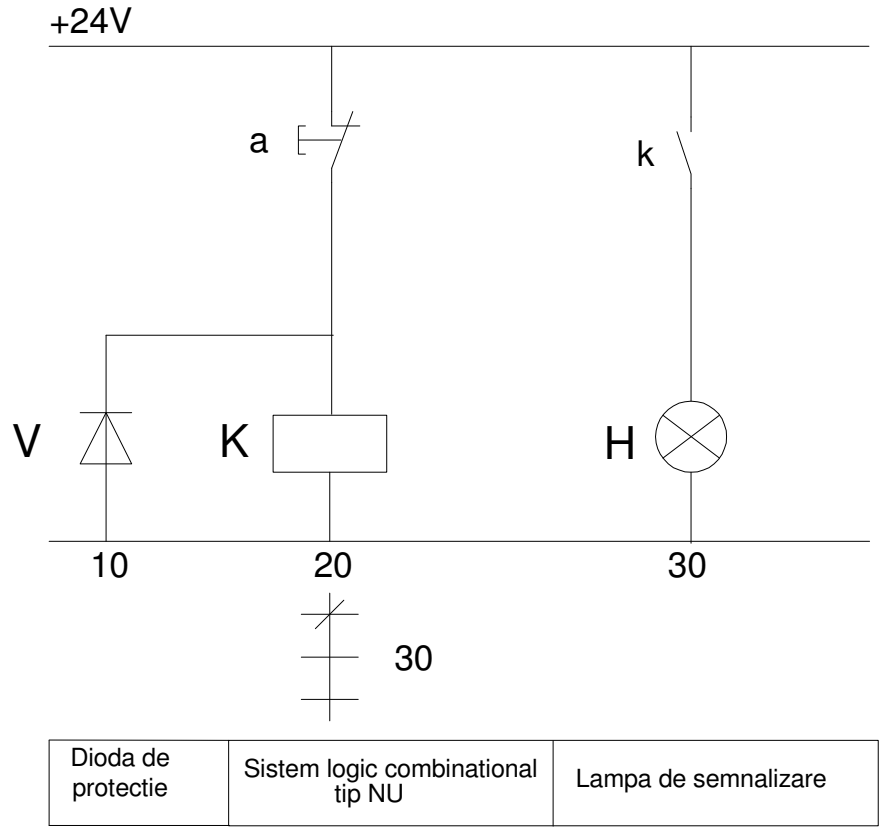


Fig. 3.4 Schema desfăşurată electrică pentru SLC de semnalizare tip NU

Dioda din circuitul 10 nu face parte dintre elementele SLC. Reamintim că acestea sunt caracterizate prin două stări. Aceasta este doar o aproximare. În realitate trecerea de la o stare la alta nu se face brusc ci în cadrul unui proces tranzitoriu care poate avea efecte nedorite. Acesta este cazul sarcinii inductive formate din bovina releului din circuitul 20. La deschiderea contactului *a* are loc un fenomen de autoinducție care produce o tensiune foarte mare care se opune tensiunii de alimentare. această tensiune provoacă scântei la contactul *a* care produc paraziți şi deteriorează contactu.

Tab. 3.7 Tabelul de adevăr cu rezultatele analizei SLC tip NU

a	***K***
0	**?….1**
1	**?….0**

Relația logică intrare – ieşire cu semnale de tip contact este:

$$k = \bar{a} \tag{3.8}$$

3.2.4. Relații logice caracteristice sistemelor logice combinaționale

Deoarece modelul teoretic al unui sistem logic combinațional este o algebră booleană între elementele sale componente există următoarele relații logice care pot fi verificate prin simulare.

- ✓ Comutativitate

$$ab = ba \qquad a + b = b + a \tag{3.9}$$

- ✓ Asociativitate

$$a(bc) = (ab)c \qquad a + (b + c) = (a + b) + c \tag{3.10}$$

- ✓ Distributivitate

$$a(b + c) = ab + ac \qquad a + bc = (a + b)(a + c) \tag{3.11}$$

- ✓ Absorbție

$$a + ab = a \qquad a(a + b) = a \tag{3.12}$$

- ✓ Idempotență

$$aa = a \qquad a + a = a \tag{3.13}$$

- ✓ Dublă negați

$$\bar{\bar{a}} = a \tag{3.14}$$

- ✓ Teorema lui De Morgan

$$\overline{ab} = \bar{a} + \bar{b} \qquad \overline{a + b} = \bar{a}\bar{b} \tag{3.15}$$

- ✓ Terțul exclus

$$a + \bar{a} = 1 \qquad a\bar{a} = 0 \tag{3.16}$$

- ✓ Relații cu constante logice

$$a0 = 0 \qquad a + 0 = a \tag{3.17}$$

$$a1 = a \qquad a + 1 = 1 \tag{3.18}$$

$$\bar{1} = 0 \qquad \bar{0} = 1 \tag{3.19}$$

3.3. Sinteza sistemelor logice combinaționale

Sinteza determinată de (3.2) este una dintre metodele de proiectare ale sistemelor logice combinaționale.

3.3.1. Metoda formei disjunctive canonice

Să considerăm sistemul logic combinațional pentru aprinderea unei lămpi de la capetele unui coridor, Fig. 3.5.

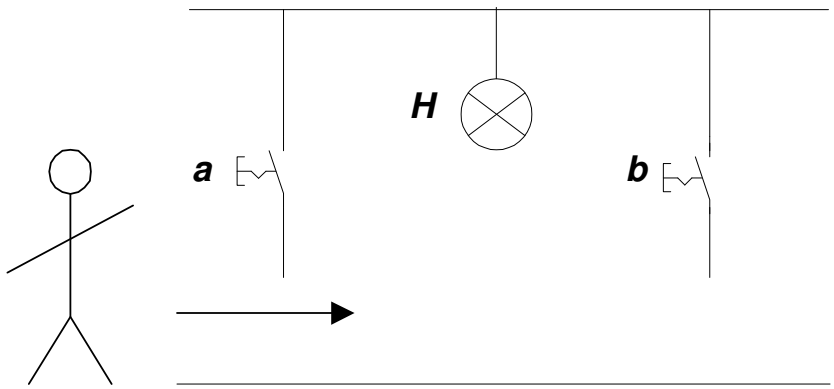


Fig. 3.5 Schema tehnologică pentru aprinderea unei lămpi de la capetele unui coridor

Semnalele de intrare în SLC sunt variabilele binare ataşate butoanelor *a* şi *b* cu memorie mecanică iar semnalul de ieşire este variabila binară ataşată lămpii *H*. Modul dorit de funcționare a SLC este prezentat în Tab. 3.8. Dacă omul nu se găseşte în coridor şi butoanele *a, b* nu sunt apăsate, Fig. 3.5, atunci lampa *H* este stinsă, situație specificată în prima linie a tabelului de adevăr Tab. 3.8.

Tab. 3.8 Tabelul de adevăr pentru SLC tip SAU-EXCLUSIV

a	***b***	***H=f(a,b)***
0	0	**f(0,0)=0**

0	1	**f(0,1)=1**
1	0	**f(1,0)=1**
1	1	**f(1,1)=0**

La intrarea în coridor omul acționează butonul *a* cu memorie mecanică. Lampa *H* trebuie să se aprindă, caz prezentat în linia trei a Tab. 3.8. Omul avansează pe coridor dar butonul *a* rămâne acționat deoarece a memorat impulsul de acționare. La capătul coridorului omul acționează prin apăsare butonul *b* care rămâne în această stare deoarece are memorie. Lampa se stinge. Situația este specificată în linia patru tabelului de adevăr Tab. 3.8. La întoarcere situația se repetă. La început ambele butoane sunt apăsate şi lampa stinsă (linia patra) La intrarea în coridor se acționează butonul *b*. Acesta fiind în starea de memorare a acționării precedente este apăsat. Prin noua acționare nu mai este apăsat iar lampa trebuie să se aprindă, situație reflectată în linia a doua a Tab. 3.8. În continuare valorile se repetă.

Cunoaştem acum intrările şi ieşirea dar nu ştim funcția logică *f* care le leagă. Pentru a o afla presupunem că relația logică este de forma:

$$H = f(a,b) = ua + v\bar{a} \tag{3.20}$$

în care *u* şi *v* sunt două funcții logice numai de variabila *b*.

Dar variabila *a* poate lua numai două valori, 0 sau 1. Să examinăm cele două cazuri pe rând.

$$a = 0 \qquad H = f(0,b) = u \cdot 0 + v \cdot 1 = v \tag{3.21}$$

$$a = 1 \qquad H = f(1,b) = u \cdot 1 + v \cdot 0 = u \tag{3.22}$$

Deci (3.20) devine:

$$H = f(a,b) = f(1,b)a + f(0,b)\bar{a} \tag{3.23}$$

Repetând procedeul pentru *f(1,b)* şi *f(0,b)* se obține *forma disjunctivă canonică* a relației logice:

$$H = f(a,b) = f(0,0)\bar{a}\bar{b} + f(0,1)\bar{a}b + f(1,0)a\bar{b} + f(1,1)ab \tag{3.24}$$

Se observă că coeficienții variabilelor *a* şi *b* sunt tocmai valorile lui *H* din Tab. 3.8. Înlocuind se obține:

$$H = \bar{a}b + a\bar{b} \tag{3.25}$$

Această relație poate fi implementată foarte uşor dacă se ține seama că funcția *SAU* înseamnă conectarea butoanelor în paralel, funcția *ŞI* înseamnă conectarea butoanelor în serie, iar funcția *NU* se realizează cu un buton având contactul normal închis. Proiectarea SLC

trebuie să țină seama însă şi de alte considerente, nu numai relația logică între intrări şi ieşiri. Dacă, de exemplu, se doreşte ca curentul prin butoane să fie mult mai mic decât curentul de sarcină al lămpii atunci se foloseşte un releu intermediar şi de obține schema din *Fig. 3.6*. Analiza acestui SLC arată că este format din două subsisteme. Primul este format din bobina releului şi butoanele din circuitele 10 şi 20 iar relația logică dintre intrări şi ieşiri (3.26) este de tipul (3.25) realizând funcția tip *SAU-EXCLUSIV*. Al doilea subsistem este format din lampa şi contactul *k* al releului care se găsesc în circuitul 30. Relația logică a acestui subsistem este (3.27) care adăugată la relația fizică (3.28) permite realizarea relației dorite (3.25) după eliminarea variabilelor intermediare *k* şi *K*.

$$K = \bar{a}b + a\bar{b} \tag{3.26}$$

$$H = k \tag{3.27}$$

$$k = K \tag{3.28}$$

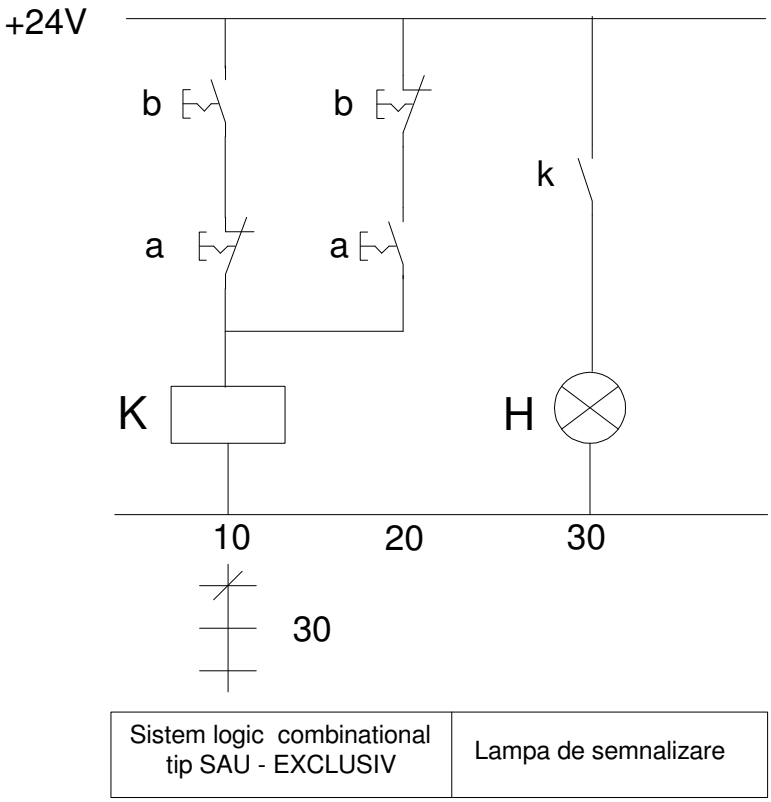


Fig. 3.6 Schema desfăşurată electrică a sistemului logic combinațional pentru aprinderea unei lămpi de la capetele unui coridor

3.3.2. Metoda diagramei Karnaugh

Sinteza sistemelor logice combinaționale prin metoda formei disjunctive canonice conduce, în general, la relații logice complexe atunci când numărul variabilelor de intrare este mare. Acest fapt poate fi ilustrat de exemplul SLC de semnalizare a majorității cu caietul de sarcini în Tab. 3.9. Lampa de semnalizare H se aprinde atunci când majoritatea butoanelor sunt acționate. Atunci când avem trei butoane de intrare majoritate este formată de două dintre ele.

Tab. 3.9 Tabelul de adevăr al SLC de semnalizare a majorității

a	***b***	***c***	***H=f(a,b,c)***
0	0	0	**f(0,0,0)=0**
0	0	1	**f(0,0,1)=0**
0	1	0	**f(0,1,0)=0**
0	1	1	**f(0,1,1)=1**
1	0	0	**f(1,0,0)=0**
1	0	1	**f(1,0,1)=1**
1	1	0	**f(1,1,0)=1**
1	1	1	**f(1,1,1)=1**

Forma disjunctivă canonică a relației logice dintre intrările şi ieşirea SLC caracterizat de Tab. 3.9 este de tipul (3.24) cu deosebirea că de data aceasta există trei semnale de intrare.

$$H = \bar{a}bc + a\bar{b}c + ab\bar{c} + abc \qquad (3.29)$$

Implementarea aceste relații cu contacte şi relee necesită 12 butoane dintre care trei sunt prevăzute cu contacte normal închise.

H

c \ a b	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Fig. 3.7 Diagrama Karnaugh pentru SLC de semnalizare a majorității

In relația (3.29) se poate da factor comun variabila a şi se obține o formă simplificată:

$$H = \bar{a}bc + a[\bar{b}c + b\bar{c} + bc] \qquad (3.30)$$

Implementarea acestei relații necesită numai 10 contacte. Alte operații caracteristice algebrei booleene ar putea produce simplificări mai importante. Există însă o alternativă care conduce la simplificarea maximă posibilă. Aceasta este metoda diagramei Karnaugh prezentată în Fig. 3.7. Diagrama Karnaugh nu este nimic altceva decât tabelul de adevăr altfel scris. Capul de tabel al diagramei Karnaugh este scris în cod Gray la care trecerea de la o valoare la alta se face prin modificarea unui singur bit. Odată ce tabelul de adevăr a fost rescris sub forma diagramei Karnaugh se realizează următoarele etape pentru obținerea relației logice.

- ✓ ***Gruparea celulelor.*** Toate celulele diagramei Karnaugh care conțin valoarea logică 1 se grupează după următoarele reguli.

R1- O grupare poate conține un număr N de celule care conțin valoarea 1 care este egal cu $N=2^n$, în care n = 0,1,2,3,4,5…Practic o grupare va conține 1, 2, 4, 8, 16 celule. Se observă că o grupare poate avea o singură celulă. Grupările se numerotează aşa cum s-a procedat în Fig. 3.7.

R2 – Grupările trebuie să fie cât mai mari şi cât mai puține.

R3 – Toate celulele dintr-o grupare trebuie să aibă câte o latură comună, nu un nod comun. Din această cauză grupările au forma unor dreptunghiuri sau pătrate şi nu a unei stele

R4 – O celulă poate să facă parte din mai multe grupări.

R5 – Suprafața în care se găsesc celulele diagramei Karnaugh ar laturile opuse lipite. În felul acesta se pot face grupări cu celulele de graniță sau celulele situate la colțuri.

- ✓ ***Relația logică cu variabile ondulate.*** Pentru fiecare grupare se scrie produsul variabilelor ondulate. Aceste produse se adună pentru a obține variabila de ieşire. Pentru Fig. 3.7 se obține:

$$H = \underline{\tilde{a}\,\tilde{b}\,\tilde{c}} + \underline{\underline{\tilde{a}\,\tilde{b}\,\tilde{c}}} + \underline{\underline{\underline{\tilde{a}\,\tilde{b}\,\tilde{c}}}} \qquad (3.31)$$

- ✓ ***Calculul variabilelor ondulate.*** Să considerăm o variabilă ondulată oarecare, să zicem ***ã***. Pentru gruparea considerată se observă ce valori are variabila neondulată ***a*** pentru fiecare dintre celulele grupării. Există trei cazuri. În primul rând este posibil ca variabila neondulată să aibă pentru toate celulele grupării valoarea unu. Atunci variabila ondulată este egală cu variabila neondulată.

$$a = 1 \quad \Rightarrow \quad \tilde{a} = a \qquad (3.32)$$

Asemănător

$$a = 0 \quad \Rightarrow \quad \tilde{a} = \bar{a} \qquad (3.33)$$

Dacă pentru o grupare variabila neondulată are pentru unele celule valoarea unu ***şi*** pentru alte celule valoarea zero atunci variabila ondulată este o constantă logică egală cu unu:

$$[a=1]\bigcap[a=0]\Rightarrow \tilde{a}=1 \qquad (3.34)$$

Aplicând aceste reguli la diagrama Karnaugh din Fig. 3.7 se obține relația logică dorită:

$$H = ab + ac + bc \qquad (3.35)$$

Dacă se dă factor comun o variabilă, de exemplu a, se obține o nouă simplificare:

$$H = a(b + c) + bc \qquad (3.36)$$

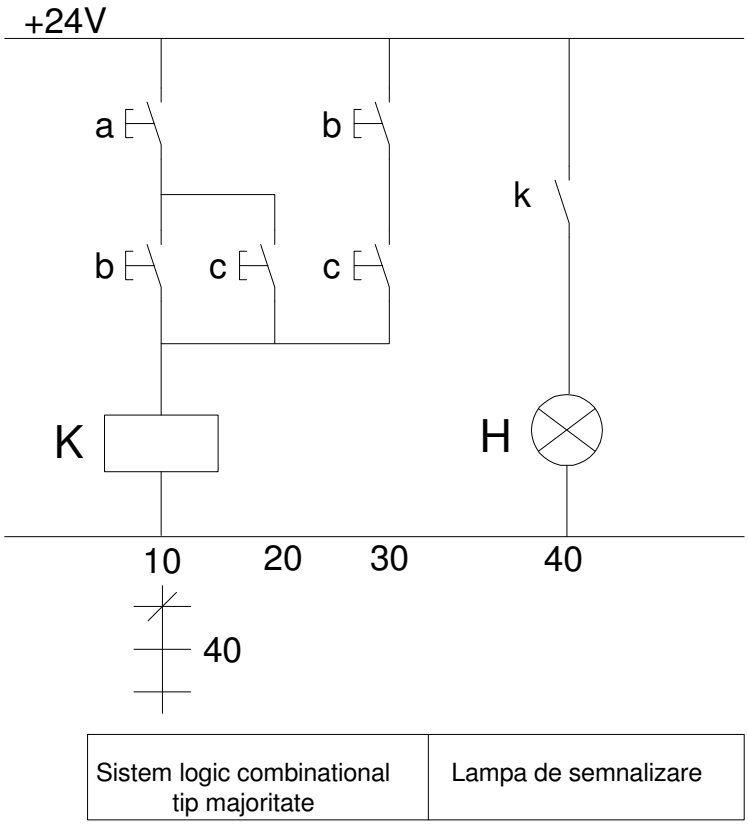


Fig. 3.8 Schema desfăşurată electrică pentru SLC cu majoritate

Implementarea relației logice (3.36) folosind un releu intermediar pentru a obține curenți mici prin contacte este prezentată în Fig. 3.8. În acest exemplu folosirea metodei de sinteză a diagramei Karnaugh a permis simplificarea schemei, comparativ cu cea obținuta la metoda formei disjunctive canonice, de la 12 la 5 contacte.

Exemplul prece3dent a arătat că sinteza SLC necesită cunoaşterea tuturor intrărilor şi ieşirilor corespunzătoare. În practică multe valori ale intrărilor nu pot apare din motive tehnologice sau ieşirile pentru ele sunt indiferente. Chiar dacă am ştii toate valorile intrărilor nu este posibilă prelucrarea lor practică deoarece numărul lor creşte exponenţial cu numărul semnalelor de intrare. Pentru n semnale de intrare avem $N=2^n$ valori ale semnalelor de ieşire. O manieră de a depăşi aceste dificultăți este prezentată în exemplul următor.

t

de \ bc	**00**	**01**	**11**	**10**
00	0	0	—	0
01	0	0	—	1
11	0	1	—	1
10	0	0	—	1

Fig. 3.9 Diagrama Karnaugh a SLC pentru semnalizarea suprasarcinii

Se doreşte sinteza unui sistem logic combinaţional care să semnalizeze apariţia suprasarcinii unui generator de 100 kW. Sarcina generatorului este formată din cinci motoare: motorul *a* de 51 kW, motorul *b* de 40 de kW, motorul *c* de 20 kW, motorul *d* de 20 kW şi motorul *e* de 10 kW. Motoarele *b* şi *c* nu funcționează niciodată simultan din motive tehnologice. Sistemul având cinci semnale logice de intrare *a,b,c,d* şi *e,* numărul valorilor posibile este de $2^5=32$. Putem evita considerarea tuturor acestor valori dacă descompunem sistemul format din cele cinci motoare în două subsisteme: motorul *a* şi grupul motoarelor *b,c,d,e*. Se observă că suprasarcina, adică consumul mai mare de 100kW, apare numai dacă motorul *a* funcționează *ŞI* al doilea subsistem consumă mai mult de 50 kW.

Deci lampa de semnalizare H se aprinde dacă:

$$H = a \cdot t \tag{3.37}$$

Să notăm cu *t* condiția ca al doilea subsistem să consume mai mult de 50 kW. Diagrama Karnaugh pentru variabila de ieşire *t* a subsistemului format din cele patru motoare este prezentată în Fig. 3.9. Relația logică corespunzătoare este:

$$t = \tilde{b}\,c\,\tilde{d}\,\tilde{e} + \tilde{b}\,c\,d\,\tilde{e} + b\,\tilde{c}\,\tilde{d}\,\tilde{e} = b11e + b1d1 + 1cde = b(e+d) + cde \tag{3.38}$$

Considerând (3.37) rezultă relația logică pentru ieşirea SLC formată din lampa H.

$$H = at = a[b(e+d)+cde] \tag{3.39}$$

Dacă se doreşte semnalizarea funcționării normale a generatorului electric, fără suprasarcină, atunci lampa H1 se va aprinde atunci când *NU* se aprinde H. Din (3.37) se obține:

$$H1 = \overline{H} = \overline{a \cdot t} = \bar{a} + \bar{t} \tag{3.40}$$

Diagrama Karnaugh pentru complementul lui *t* se obține din Fig. 3.9 în care se iau toate valorile negate ca în Fig. 3.10.

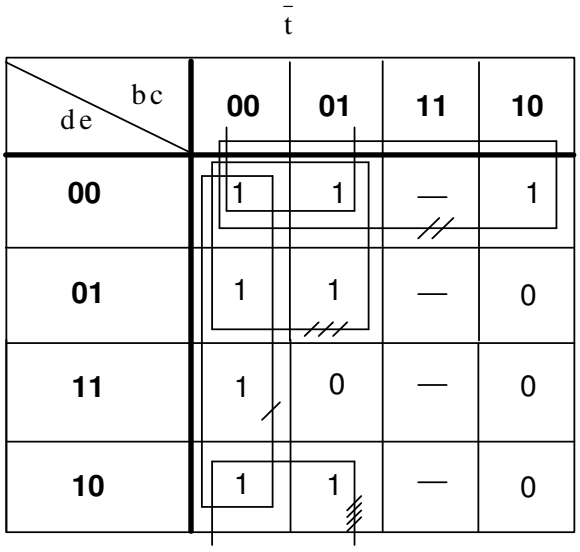


Fig. 3.10 Diagrama Karnaugh a SLC de semnalizare a funcționării fără suprasarcină a generatorului

După ce se fac grupările şi se calculează variabilele ondulate se obțin e relația:

$$\bar{t} = \bar{b}\bar{c} + \bar{d}\bar{e} + \bar{b}\bar{d} + \bar{b}\bar{e} \tag{3.41}$$

Din (3.40) rezultă condiția de aprindere a lămpii H1 la funcționarea normală a generatorului:

$$H1 = \bar{a} + \bar{t} = \bar{a} + \bar{b}(\bar{c}+\bar{e}) + \bar{d}(\bar{e}+\bar{b}) \tag{3.42}$$

Relația (3.41) se poate obține şi din (3.38) dacă se aplică teorema lui De Morgan şi se consideră relația (3.18).

$$\bar{t} = \overline{be+bd+cde} = (\overline{be})(\overline{bd})(\overline{cde}) = (\bar{b}+\bar{e})(\bar{b}+\bar{d})(\bar{c}+\bar{d}+\bar{e}) \tag{3.43}$$

$$\bar{t} = (\bar{b}+\bar{b}\bar{d}+\bar{e}\bar{b}+\bar{e}\bar{d})(\bar{c}+\bar{d}+\bar{e}) = \bar{b}\bar{c}+\bar{d}\bar{e}+\bar{b}\bar{d}+\bar{b}\bar{e} \tag{3.44}$$

3.4. Testarea sistemelor logice combinaționale

Problema testării a fost definită de (3.3). Dacă dorim să testăm SLC din Fig. 3.11 trebuie să cunoaştem o valoare a ieşirii. În cazul de față dorim ca lampa să fie aprinsă, deci Z=1. Pentru a testa dacă SLC aprinde corect lampa este nevoie să ştim care sunt semnalele de intrare pentru care lampa este aprinsă. Aceasta se rezolvă uşor dacă cunoaştem relația logică dintre intrări şi ieşire. Deoarece de obicei se cunoaşte schema desfăşurată electrică, ca în Fig. 3.11, se face mai întâi o analiză SLC din care rezultă (3.45).

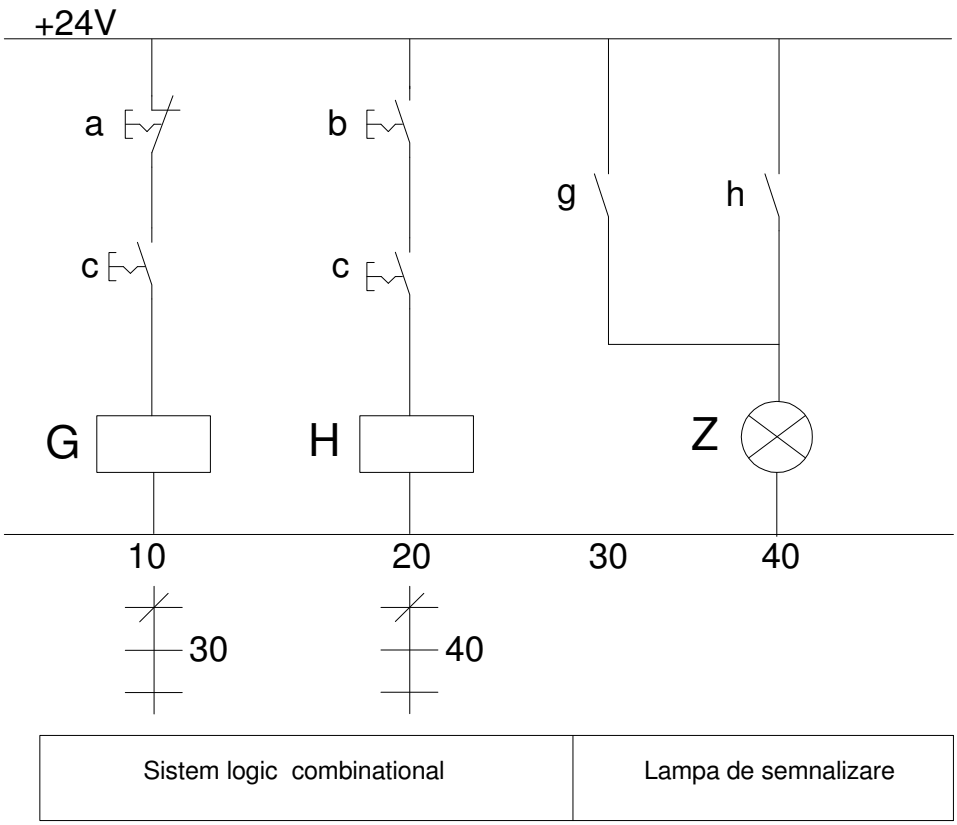


Fig. 3.11 Sistem logic combinațional supus testării

$$Z = g + h = \bar{a}c + bc = c(\bar{a} + c) \qquad (3.45)$$

Cu ajutorul relației logice intrări – ieşire se determină tabelul de adevăr prezentat în Tab. 3.10. Din acesta se constată care sunt intrările pentru care ieşirea are valoarea unu, adică lampa este aprinsă. Acestea sunt

$$(a, b, c) = (0,0,1) \qquad (3.46)$$

$$(a,b,c) = (0,1,1) \tag{3.47}$$

$$(a,b,c) = (1,1,1) \tag{3.48}$$

Tab. 3.10 Tabelul de adevăr pentru SLC din Fig. 3.11 care este testat

a	***b***	***c***	***Z***
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**0**
1	1	0	**0**
1	1	1	**1**

Relațiile (3.46), (3.47) şi (3.48) spun că sistemul logic combinațional din Fig. 3.11 poate fi testat dacă este defect prin apăsarea butonului *a* sau apăsarea simultană a lui *a* şi *b* sau apăsarea simultană a tuturor celor trei butoane *a,b,c*. Această metodă de testare pune în evidență faptul că a apărut o defecțiune atunci când lampa nu se aprinde, dar nu oferă nici o indicație asupra locului unde a apărut defecțiunea.

Examinând manevra (3.46) se constată că la apăsarea butonului *c* lampa Z se aprinde prin intermediul releului G. Dacă lampa nu se aprinde înseamnă că releul este stricat şi am localiyat defecțiunea în circuitul cu numărul 10. În mod asemănător rezultă că dacă lampa Z nu se aprinde la apăsarea tuturor celor trei butoane releul H din circuitul 20 este defect. Semnalul de intrare (3.29) nu permite localizarea defecțiunii deoarece lampa se poate aprinde sau prin releul G sau prin releul H. Dacă unul dintre relee este defectat lampa tot se prind prin intermediul celuilalt releu şi defecțiunea nu este pusă în evidență.

La sistemele logice combinaționale realizate cu contacte şi relee tipurile cele mai frecvente de defecțiuni îl constitue scurtcircuitele la masă sau la borna de alimentare care conduc la înțepenirea unor relee pe valoarea acționat sau valoarea neacționat. Depistarea tipului de defect se poate face cu ajutorul relațiilor logice diferențiale.

3.5. Implementarea SLC cu automate programabile logice

Schema desfăşurată electrică a automatului implementat cu contacte şi relee foate fi transformată uşor într-un program pentru APL În Fig. 3.12 şi Fig. 3.13se prezintă programarea funcţiilor Şi, SAU, NU sub formă de schemă desfăşurată (ladder diagram) folosind simbolurile din standardele european şi american.

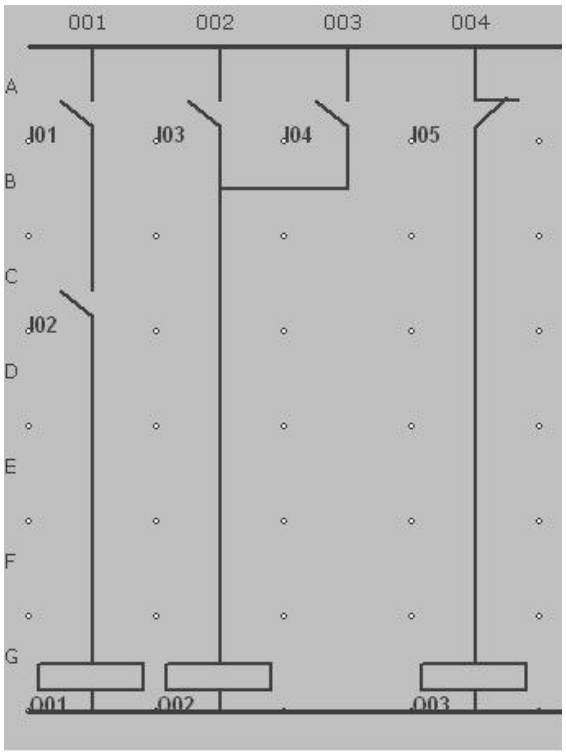


Fig. 3.12 Programul APL pentru fubcţiile ŞI, SAU, NU sub formă de schemă desfăşurată (LD) conform standardului DIN/IEC

Schema desfăşurată electrică a APL din Fig. 3.14 specifică şi configurarea automatului, adică corespondenţa dintre semnalele de intrare / ieşire şi denumirile corespunzătoare ale bornelor APL.

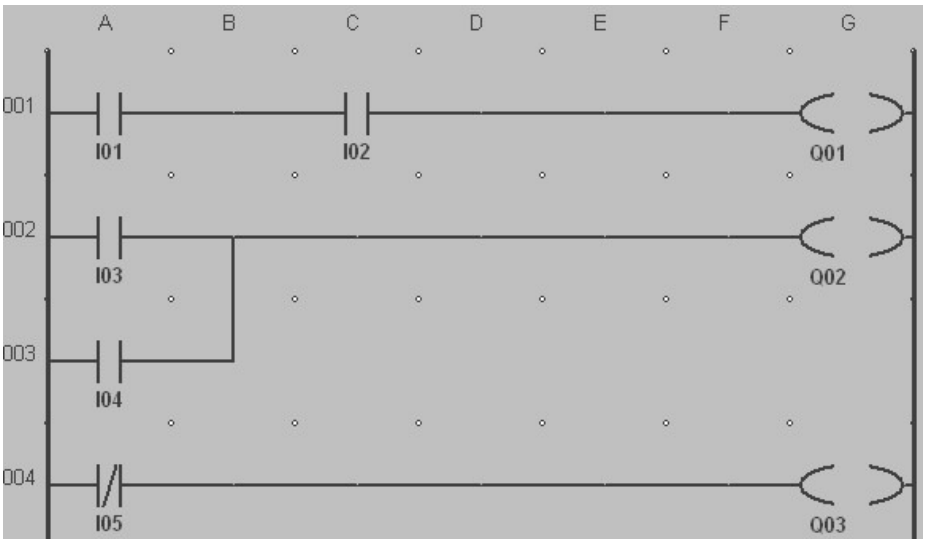


Fig. 3.13 Programul APL pentru fubcțiile ŞI, SAU, NU sub formă de schemă desfăşurată (LD) conform standardului ANSI-CSA

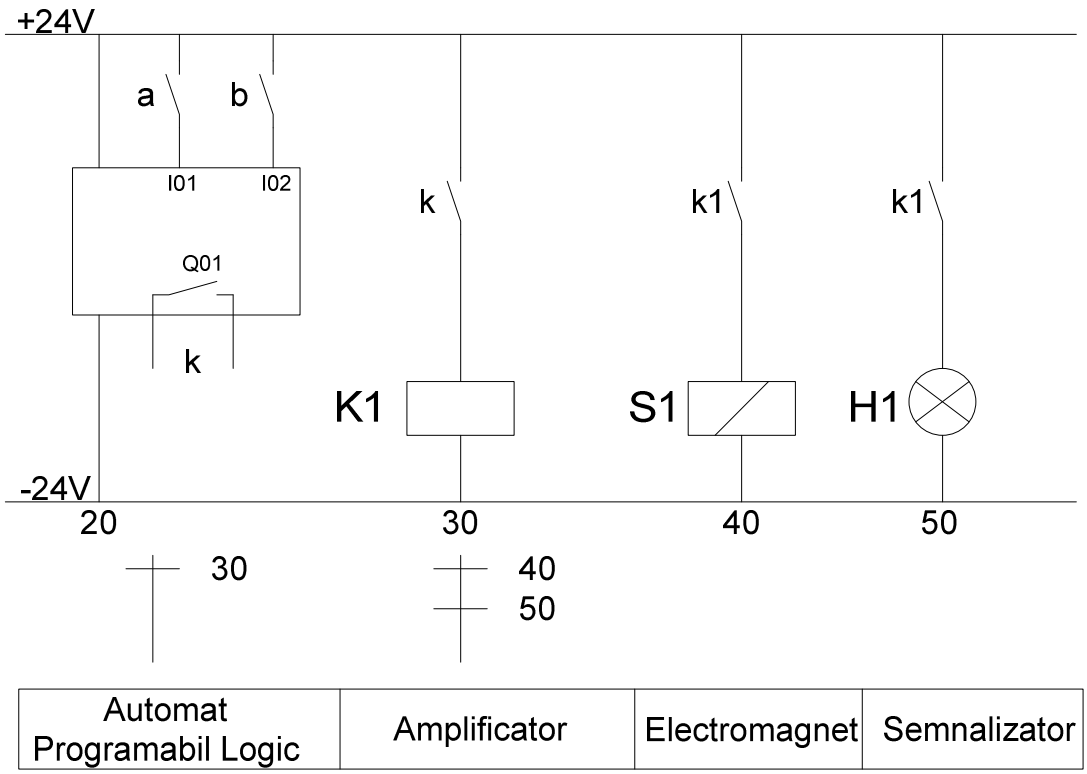


Fig. 3.14 Schema desfăşurată electrică pentru un APL cu specificarea intrărilor şi ieşirilor.

3.6. Implementarea SLC cu circuite electronice integrate

Funcţiile logice pot fi implementate cu ajutorul circuitelor logice integrate. Simbolurile acestor circuite sunt prezentate în Fig. 3.15. Un exemplu de implementare îl prezintă filtrul logic pentru ascensor.Ieşirea filtrului ia valoarea 1 atunci când greutatea G din cabina ascensorului se găseşte între anumite limite. Pentru aceasta în cabină se montează trei traductori de greutate a, b şi c a căror ieşiri iau următoarele valori:

- a = 1 dacă G > 0 kg
- b = 1 dacă G ≥ 100 kg
- c = 1 dacă G ≥ 800 kg

trbuie să funcţioneze numai atunci când cabina este goală, adică G = 0 kg, sau atunci când greutatea G este cuprinsă între 100 şi 800 kg. Tabelul de adevăr este prezentat în Tab. 3.11 iar diagrama Karnaugh în Fig. 3.16.

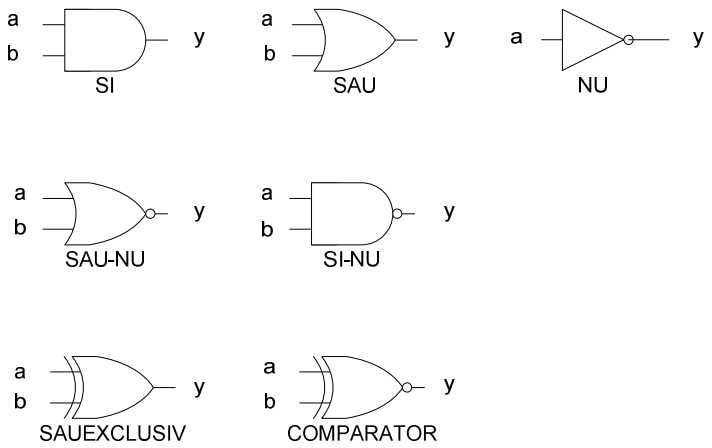


Fig. 3.15 Simbolurile funcţiilor logice implementate cu circuite electronice integrate.

Tab. 3.11

a	***b***	***c***	***y***
0	0	0	**1**
0	0	1	**-**
0	1	0	**-**
0	1	1	**-**
1	0	0	**0**
1	0	1	**-**
1	1	0	**1**
1	1	1	**0**

y

c \ a b	00	01	11	10
0	1	-	1	0
1	-	-	0	-

Fig. 3.16 Diagrama Karnaugh pentru filtrul logic al ascensorului

Se fac grupările şi rezultă funcția logică:

$$y = \underline{\tilde{a}\,\tilde{b}\,\tilde{c}} + \underline{\underline{\tilde{a}\,\tilde{b}\,\tilde{c}}} \tag{3.49}$$

$$y = \bar{a} + b\bar{c} \tag{3.50}$$

Schema corespunzătoare este prezentată în Fig. 3.17

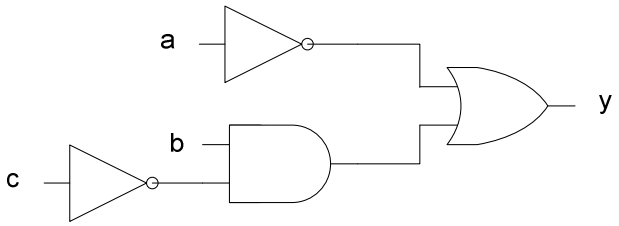


Fig. 3.17 Filtrul logic al ascensorului implementat cu circuite ŞI-SAU-NU.

Relația (3.50) şi i se poate aplica teorema lui De Morgan:

$$\bar{y} = \overline{\bar{a} + b\bar{c}} = \bar{\bar{a}} \cdot \left(\overline{b \cdot \bar{c}}\right) \tag{3.51}$$

Dacă se mai neagă odată se obține:

$$y = \overline{a \cdot \left(\overline{b \cdot \bar{c}}\right)} \tag{3.52}$$

Această relație poate fi implementată numai cu circuite ŞI-NU ca în Fig. 3.18.

3.7. Aspecte constructive ale sistemelor logice combinaționale

Un sistem logic combinațional poate fi realizat constructiv (implementat) în mai multe variante:

- SLC electric cu contacte şi relee,

- SLC electronic cu circuite integrate ŞI, SAU, NU,
- SLC electronic cu automat programabil logic – APL
- SLC electronic cu microcalculator de proces,
- SLC hidraulic,
- SLC pneumatic.

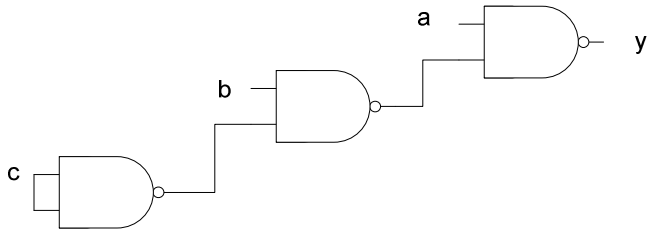


Fig. 3.18 Filtrul logic al ascensorului implementat cu circuite ŞI-NU

3.7.1. Construcția releului electromagnetic

Releul este un echipament electromecanic a cărui construcție este reprezentată schematic în Fig. 3.19 şi Fig. 3.20. El este format dintr-o bobină (1), o armătură (2) mobilă legată de resortul elastic (3) şi două contacte (4).

Atunci când prin bobină trece curent releul este acționat şi armătura sa este atrasă trcând de pe borna superioară a contactului pe borna sa inferioară. Contactul (4) este un contact comutator. Releele pot fi dotate constructiv în afară de contacte comutatoare şi cu contacte normal închise sau normal deschise[76]. În situația în care curentul nu mai trece prin bobină resortul comută armătura pe borna superioară a contactului.

Din motive constructive[77] folosirea releelor şi contactelor îndeplinesc, în general, următoarele condiții:

- O bornă a bobinei se leagă întotdeauna la aceiaşi bornă comună a sursei de alimentare.
- La cealaltă bornă a sursei de alimentare se leagă, pe cât posibil, contactele normal închise cu funcție de protecție.

---

[76] În Fig. 3.19 armătura şi borna superioară formează un contact normal închis, iar armătura şi borna inferioară formează un contact normal deschis. Se observă că contactul comutator este format dintr-un contact normal deschis şi unul normal închis. Adeseori contactele comutatoare ale releelor îndeplinesc numai funcția de contact normal deschis sau contact normal închis.

[77] Prin motive constructive înțeleg o gamă larga de situații legate de tehnologie, fiabilitate, siguranță în funcționare, protecția omului şi a aparatelor şi simplitatea exploatării.

- La o bornă a bobinei sau contactului se leagă întotdeauna nu mai mult de două fire.
- În paralel pe bobină se montează un dispozitiv de antiparazitare care stinge scânteile ce apar la deschiderea contactelor releului.

În cadrul unui sistem logic combinațional releul poate îndeplinii mai multe funcțiuni, realizarea lui constructivă dobândind caracteristici specifice.

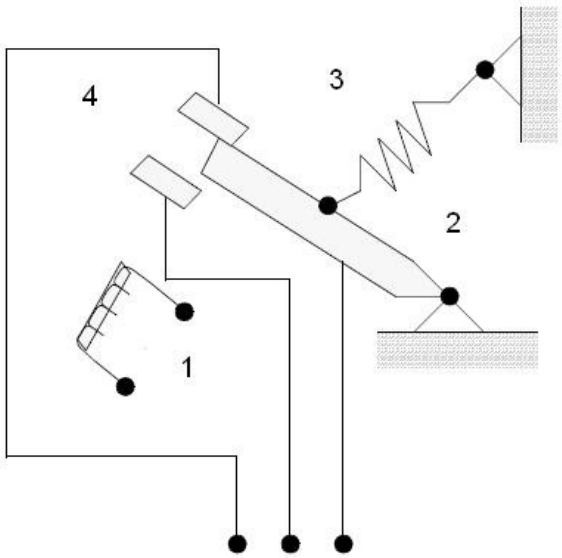


Fig. 3.19 Principiul de funcționare al unui releu

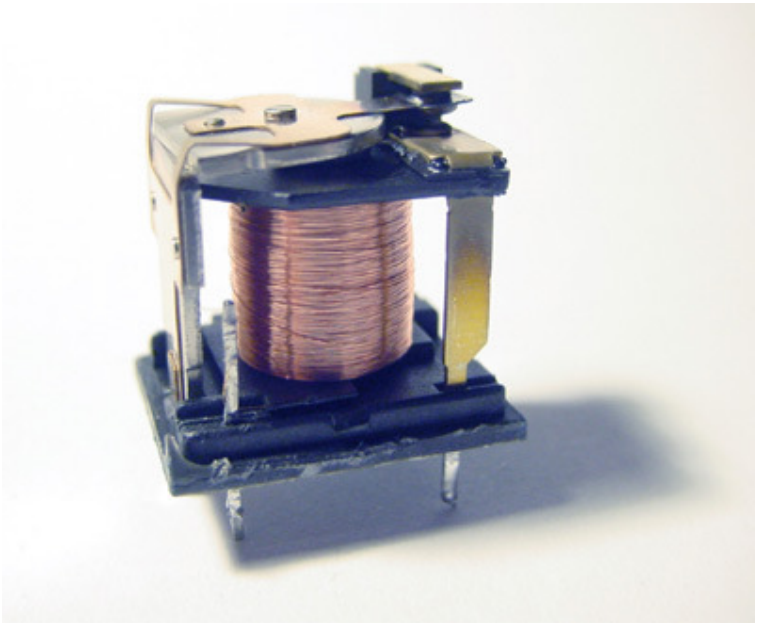


Fig. 3.20 Construcția unui releu pentru automatizare

3.7.2. Elemente reale ale sistemelor logice combinaționale

Elementele SLC sunt aproximate prin modele teoretice ideale care au numai două stări. În realitate lucrurile nu stau aşa. De exemplu contactele sunt considerate drept elemente care lasă să treacă sau întrerup total curentul electric. Contactul real are însă o rezistență electrică cu o valoare cuprinsă între 0,01 şi 10 Ω. Valorile mari ale rezistenței de contact sunt atinse după o funcționare mai îndelungată şi conduc, în cazul unor curenți mari, la căderi de tensiune importante care periclitează acționarea releului. Acest fenomen limitează numărul contactelor conectate în serie, fenomen care nu este luat în considerare de modelul teoretic sub formă de algebră booleană al sistemelor logice combinaționale.

Caracteristica statică de acționare a releului nu este nici ea simplă. Forma ei este de tip histerezis şi trebuie luată în considerare la proiectarea sistemelor logice combinaționale.

La deschiderea contactelor se produc scântei care produc perturbații pentru aparatura electronică şi calculatoare şi conduc la deteriorarea mai rapidă a contactelor. Eliminarea acestui fenomen se numeşte deparazitare şi se face prin conectarea în paralel cu bobinele releelor a unor dispozitive.

Pentru releele alimentate în curen continuu se montează diode conectate invers care nu conduc în mod normal dar scurtcircuitează tensiunile electromotoare induse la deschiderea contactelor.

Pentru releele alimentate în curent alternativ se folosesc varistoare sau circuite RC

3.7.3. Hazardul combinațional.

Implementarea relației logice (3.16) este prezentată în Fig. 3.21. Se observă uşor că indiferent dacă contactul a este acționat sau nu este acționat lampa H nu este aprinsă niciodată. În realitate cele două contacte, unul normal deschis şi altul normal închis, ale butonului a nu pot fi acționate simultan. Dacă contactul normal deschis este acționat înaintea contactului normal închis atunci lampa H se aprinde o fracțiune de secundă. Acest impuls de scurtă durată poate fi memorat de alte echipamente ale sistemului de conducere şi poate conduce în felul acesta la perturbări importante ale funcționării sistemului. Deoarece acționarea nesimultană ale celor două contacte este aleatotie fenomenul se numeşte hazard combinațional.

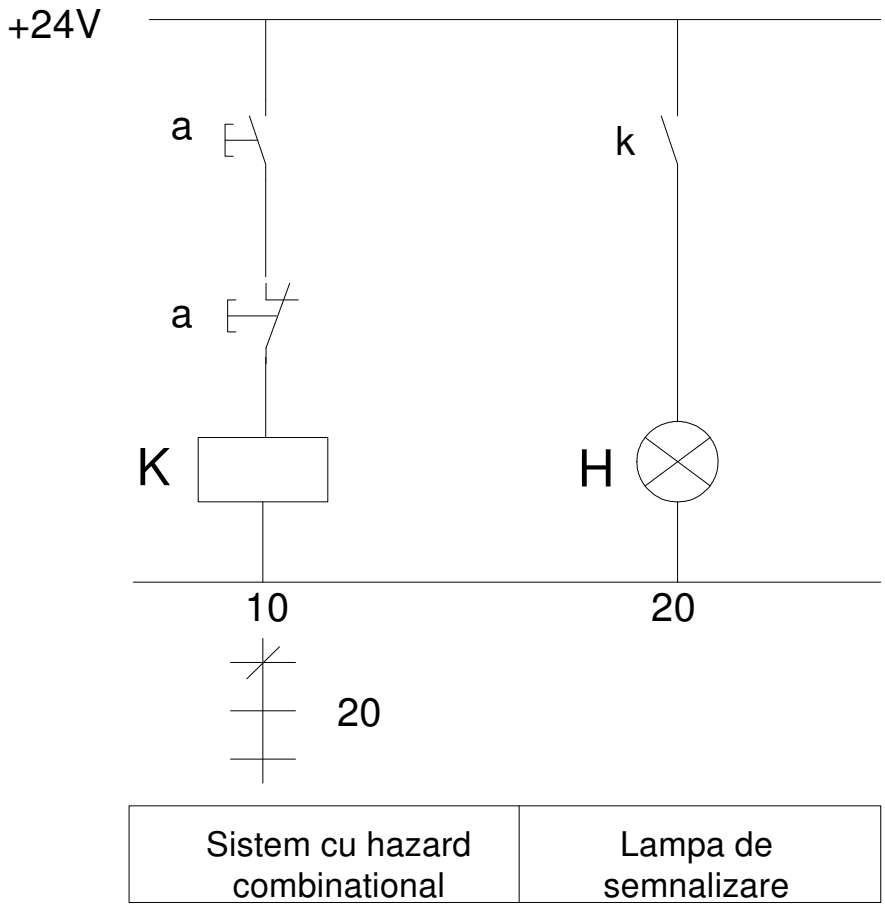


Fig. 3.21 Schema electrică desfăşurată electrică a unui SLC cu hazard combinațional

4. Sisteme automate cu evenimente discrete

Evenimentele discrete apar în mod frecvent la organizarea serviciilor oferite de clădiri şi locuinţe, în funcţionarea instalaţiilor, echipamentelor informatice şi a atelierelor de producţie. Din această cauză sistemele cu evenimente discrete sunt foarte utile în luarea deciziilor referitoare la diferite aspecte ale activităţii inginereşti, economice şi organizatorice legate de clădiri şi locuinţe. *Sistemele discrete logice* sunt cazuri particulare, mai simple, de sisteme cu evenimente discrete şi sunt folosite intensiv la conducerea automată a instalaţiilor şi echipamentelor din clădiri şi locuinţe. Ele vor fi considerate drept o *generalizare* a sistemelor logice combinaţionale[78] *SLC* sau o *particularizare* a sistemelor cu evenimente discrete[79] SED.

Trei *modele* importante folosite în analiza şi sinteza sistemelor discrete logice sunt reţeaua Petri interpretată sigură, maşina de stări şi grafcetul.

Spre deosebire de sistemele logice combinaţionale sistemele discrete logice au o comportare *dinamică,* adică semnalele lor de intrare şi ieşire depind de timp, şi au *memorie* pentru că semnalele lor de ieşire depind de starea precedentă. Legătura dintre semnalele de intrare şi semnalele de ieşire *logice* ale unui astfel de sistem este prezentată adeseori *grafic* sub forma unei reţele Petri[80] interpretate sigure, a unui graf de comandă etapă-tranziţie GrafCET[81] sau a unui grafic numit maşină de stări.

Sistemele discrete logice se deosebesc, în principal, de sistemele cu evenimente discrete prin faptul că mărimile de intrare, ieşire şi stare sunt logice, adică pot avea numai două valori: 0 sau 1.

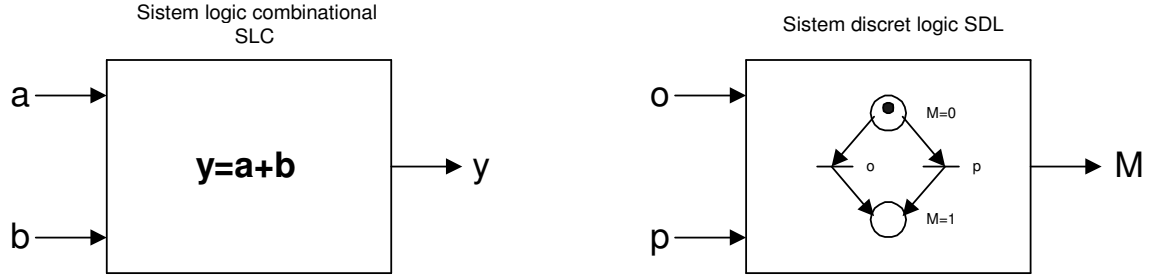


Fig. 4.1 Exemple de relaţii intrare-ieşire la un SLC şi la un SDL

La sistemele logice combinaţionale relaţia intrare-ieşire poate fi reprezentată sub forma unui tabel de adevăr sau a unei funcţii logice, ca în *Fig. 4.1* a).

[78] Sisteme logice combinaţionale generalizate prin introducerea unei reacţii (feedback).

[79] Sisteme cu evenimente discrete particularizate prin admiterea numai a valorilor logice pentru stări.

[80] Concept elaborată de Carl Adam Petri în 1962 în Germania.

[81] Concept elaborat de o comisie în 1979 în Franţa.

La sistemele discrete logice aceiaşi relaţie intrare – ieşire poate fi reprezentată ca în *Fig. 4.1* b) de un graf, o reţea Petri interpretată sigură în acest exemplu, sau de un sistem de funcţii logice de timp.

Sistemele cu evenimente discrete pot realiza câteva operaţii importante pentru conducerea sistemelor[82]: secvenţialitatea, paralelismul general, paralelismul conveier, concurenţa, interblocarea şi sincronismul.

Problemele sistemelor logice combinaţionale, analiza, sinteza (proiectarea), testarea, optimizarea şi implementarea se regăsesc şi la sistemele cu evenimente discrete.

Analiza sistemelor discrete logice se face pornind de la faptul că acestea sunt cazuri particulare ale sistemelor cu evenimente discrete. Se distinge analiza structurală de analiza comportamentală a acestor sisteme.

In esenţă, proiectarea sistemelor discrete logice folosite în conducerea instalaţiilor şi proceselor din clădiri se va baza pe proiectarea sistemelor logice combinaţionale.

Implementarea modelelor stabilite pentru sistemele discrete logice este mai dificilă decât în cazul sistemelor logice combinaţionale. Se va pune accentul pe implementarea cu contacte şi relee şi implementarea cu automate programabile logice.

4.1. Modelarea sistemelor cu evenimente discrete

Sistemele discrete logice pot fi considerate drept cazuri particulare de sisteme cu evenimente discrete şi din această cauză vom studia, la început, cazul, mai general, al modelării sistemelor cu evenimente discrete.

Există multe modele ale sistemelor cu evenimente discrete[83]. In domeniul sistemelor cu conducere automată se folosesc drept modele foarte mult *reţelele Petri interpretate sigure*[84] care sunt nişte cazuri mai simple, de reţele Petri. Vom folosi acest tip de modele atât pentru sistemul automat[85], în întregul său, cât şi pentru partea din sistemul automat care este formată din echipamentul de conducere.

In *Fig. 4.2* se prezintă reţeaua Petri interpretată care modelează sub forma unui sistem cu evenimente discrete SED funcţionarea unei instalaţii electrice formată din trei resurse: echipamentul de alimentare al motorului, echipamentul de alimentare al automatului motorului şi motorul electric. Intr-o primă aproximaţie[86] se disting trei stări in funcţionarea acestui SED: nealimentat, alimentat şi oprit, pornit. Stările sistemului sunt reprezentate în reţeaua Petri sub forma unor cercuri numite *poziţii* sau locaţii. Reţeaua Petri din *Fig. 4.2* are trei poziţii: P1, P2 şi P3. Resursele unei stări a SED sunt reprezentate sub forma unor *mărci*

[82] Instalaţiilor.

[83] Dintre aceste modele menţionăm: limbajele naturale, limbaje formale, automate stochastice temporizate, maşini de stări, reţele Petri şi grafuri de evenimente, lanţuri Markov, reţele de cozi de aşteptare, diagrame ale fluxurilor de date, etc.

[84] Se mai numesc reţele Petri interpretate logice sau reţele Petri interpretate binare..

[85] Instalaţia automatizată.

[86] Pentru motoarele de putere pornite după algoritmul stea-triunghi există mai multe stări.

sau jetoane prezente în poziție. De exemplu, poziția P1 din exemplul considerat are două resurse: echipamentul de alimentare al automatului şi echipamentul de alimentare şi protecție a motorului. O stare este activă dacă dispune de resurse, adică există mărci în interiorul cercului care o reprezintă. Dacă starea este activă se execută acțiunile care o interpretează şi sunt prezentate în dreptul poziției sale. Pentru instalația din *Fig. 4.2* prezentată în momentul inițial există o singură stare activă, poziția P1 şi se execută următoarele acțiuni: automatul şi motorul nu sunt alimentate. Trecerea de la o stare activă la altă stare activă a SED se face prin intermediul unor *arce orientate ponderate* si a unor *tranziții* reprezentate sub forma unor *dreptunghiuri înnegrite*[87]. In *Fig. 4.1* există trei tranziții: T1, T2 şi T3. Tranzițiile se *declanşează*[88] la apariția unor evenimente externe, dacă există resursele necesare, adică sunt *validate*. Evenimentele ataşate tranzițiilor T1, T2 şi T3 sunt acționarea în impuls[89] a butoanelor: *b* de alimentare, *p* de pornire şi *o* de oprire. După declanşare se consumă resurse din pozițiile precedente şi se creează resurse în pozițiile posterioare. Numărul resurselor consumate şi create este dat de ponderea arcelor.

[87] Un exemplu clasic de reprezentare a tranzițiilor este prezentat în *Fig. 4.6*. In alte figuri tranzițiile sunt reprezentate prin dreptunghiuri foarte subțiri, mai degrabă linii îngroşate, în maniera folosită de grafcet.

[88] Fire (eng).

[89] Butoanele nu au memorie mecanică.

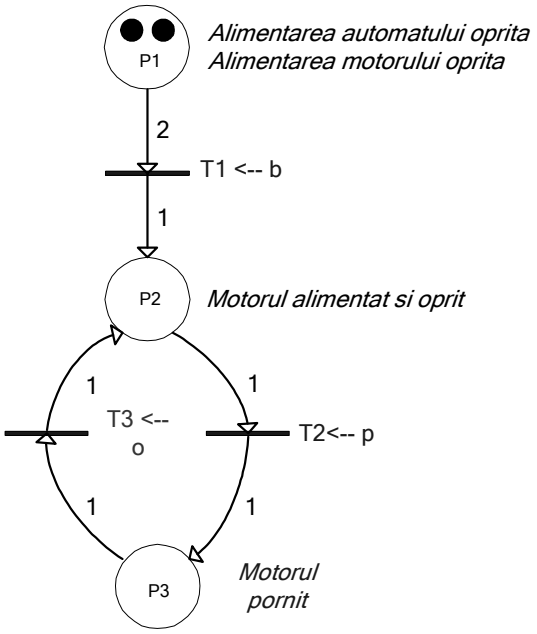


Fig. 4.2 O rețea Petri interpretată[90] *care modelează sistemul de conducere al unui motor electric.*

De exemplu, pentru situația din *Fig. 4.2* dacă apare evenimentul acționării butonului de alimentare, adică *b = 1*, tranziția T1 la care acest eveniment o interpretează se declanşează şi se consumă două mărci din poziția P1 si se creează o marcă în poziția P2. Se obține rețeaua Petri din *Fig. 4.3*. Dacă în aceiaşi situație se apasă butonul de pornire *p* sau butonul de oprire *o* tranzițiile la care sunt ataşate aceste evenimente nu se declanşează pentru că nu există resursele necesare.

O tranziție declanşabilă la apariția evenimentului asociat se numeşte *validată.* In *Fig. 4.2* este validată T1 iar în *Fig. 4.3* este validată T2. In acest din urmă caz dacă se apasă butonul de pornire *p* se declanşează T2 şi devine activă P3. Motorul este pornit. Acum în P3 se găseşte o marcă şi T3 este validată. La apăsarea butonului de oprire *o* se declanşează T3 şi se ajunge din nou în situația din *Fig. 4.3* cu P2 activă şi motorul oprit. Se observă că la apăsarea butoanelor de pornire şi de oprire pozițiile P2 şi P3 devin pe rând active şi se formează un ciclu repetitiv. Poziția P1 nu mai poate devenii nici odată activă, adică alimentarea automatului şi a motorului nu poate fi deconectată. Această situație descoperită în

[90] Rețeaua este *nesigură* pentru că în P1 sunt mai multe mărci.

urma analizei evoluției sistemului cu evenimente discrete SED descris de rețeaua Petri din *Fig. 4.3* este inacceptabilă şi trebuie remediată.

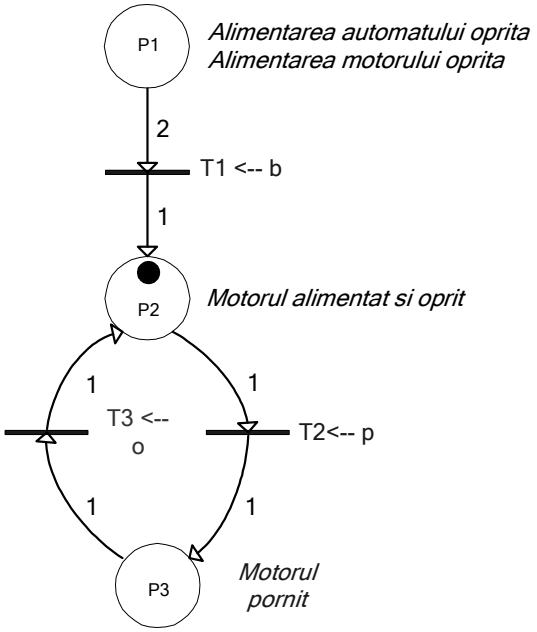


Fig. 4.3 Rețeaua Petri interpretată din Fig. 4.2 după acționarea butonului de alimentare. Poziția P2 este activă şi se execută acțiunea: motorul alimentat şi oprit.

Un fenomen interesant apare atunci când se apasă simultan pe butoanele de pornire şi oprire iar poziția P2 sau P3 este activă. Sistemul va bascula. Dacă motorul este oprit va porni şi dacă este pornit se va opri. Acest fenomen de basculare este în general inacceptabil şi se înlătură. Pentru SED complexe modelarea cu ajutorul rețelelor Petri este foarte utilă deoarece permite descoperirea şi înlăturarea diferitor defecțiuni sau fenomene nedorite.

Rețeaua Petri RP sub formă grafică este suficientă în cazurile simple. Pentru SED complexe se face analiza cu ajutorul calculatorului şi aceasta implică existența unui model matematic format din următorul 5-uplu[91]

$$RP(N, M_0) = \langle P, T; \mathrm{Pr}\, e, Post; M_0 \rangle \qquad (4.1)$$

$$N = (P, T; \mathrm{Pr}\, e, Post) \qquad (4.2)$$

[91] Cvintuplu.

în care N desemnează *structura rețelei,*

M_0: vectorul marcaj inițial al rețelei.

P: mulțimea pozițiilor,

T: mulțimea tranzițiilor,

Pre: matricea de incidență precedentă (înapoi) a tranzițiilor,

Post: matricea de incidență posterioară (înainte) a tranzițiilor,

Pentru rețeaua Petri din *Fig. 4.2* se obține:

$$P = \{P1, P2, P3\} \tag{4.3}$$

$$T = \{T1, T2, T3\} \tag{4.4}$$

Matricele Pre şi Post depind de *structura N* a rețelei. Matricea Pre se construieşte cu ajutorul tabelului următor în care se trec valorile arcelor care leagă fiecare tranziție cu poziția precedentă corespunzătoare. De exemplu, înainte de tranziția T1 se găseşte poziția P1 de care este legată cu un arc orientat cu valoarea 2.

P/t	**T1**	**T2**	**T3**
P1	2	0	0
P2	0	1	0
P3	0	0	1

Deci:

$$\Pr e(p,t) = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.5}$$

In mod asemănător rezultă matricea Post considerând valorile arcelor care leagă fiecare tranziție cu o poziție posterioară.

$$Post(p,t) = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{4.6}$$

Vectorul marcajelor inițiale M_0 este:

$$M_0(p) = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \tag{4.7}$$

Rețeaua Petri $RP(N,M_0)$ poate fi caracterizată cu ajutorul următoarei matrici de incidență C(p,t):

$$\forall p \in P, \quad \forall t \in T: \qquad C(p,t) = Post(p,t) - \Pr e(p,t) \tag{4.8}$$

Dacă se cunoaşte matricea de incidență C(p,t) atunci matricele de incidență înapoi Pre(p,t) şi înainte Post(p,t) pot fi calculate cu ajutorul următoarelor relații:

$$Post(p,t) = \max[0, C(p,t)] \tag{4.9}$$

$$\Pr e(p,t) = \max[0, -C(p,t)] \tag{4.10}$$

Pentru rețeaua Petri interpretată din *Fig. 4.2* matricea de incidență *C* rezultă din (4.8), (4.5) şi (4.6)

$$C(p,t) = Post(p,t) - \Pr e(p,t) = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} - \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -2 & 0 & 0 \\ 1 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \tag{4.11}$$

Matricea de incidență C depinde de *structura N* a rețelei Petri RP(N,M_0).

4.2. Definirea sistemelor discrete logice.

Sistemele discrete logice *sunt sisteme cu evenimente discrete* care pot fi modelate cu ajutorul unor rețele Petri *interpretate şi sigure*.

Interpretarea rețelei Petri a fost explicată pe larg în paragraful precedent. Condiția de interpretare impune ca pe lângă variabilele de stare ale sistemului, exprimate prin vectorul său de marcaj M, să existe şi variabile *logice* de intrare şi ieşire (*Fig. 4.1*). Odată cu intrările care interpretează tranzițiile se introduce şi variabila independentă *timp.* Deoarece o intrare, sub forma unui eveniment, influențează comportarea sistemului numai dacă tranziția pe care o interpretează este validată, se observă că timpul intervine numai prin valoarea sa relativă Δt parcursă de la activarea poziției conectată la tranziție. Comportarea dinamică a sistemelor discrete logice este determinată de producerea (apariția) unor *evenimente*[92] la momente discrete asincrone de timp şi nu este *antrenată de timp*[93] ca în cazul sistemelor continue sau discrete cu eşantionare.

O rețea Petri este *sigură* dacă în toate situațiile de funcționare a sistemului cu evenimente pe care îl modelează pozițiile au un număr de mărci mai mic sau egal cu unu. Asupra siguranței rețelelor Petri se va reveni în paragrafele următoare în legătură cu proprietatea de mărginire a sistemelor cu evenimente discrete. Proprietatea de siguranță este asemănătoare cu stabilitatea sistemelor continue sau discrete cu eşantionare. Ea impune de asemenea ca variabilele de stare ale sistemului să fie de tip *logic,* la fel cu intrările şi ieşirile

[92] Sistemele discrete logice sunt de tip *event – driven.*

[93] Sistemele continue sau discrete cu eşantionare sunt de tip *time – driven.*

sistemului, şi în felul acesta sistemul să poată fi implementat cu ajutorul unor dispozitive specifice, cum ar fi contactele şi releele sau automatele programabile logice.

4.3. Tipuri de rețele Petri.

Rețelele Petri sunt de tipuri variate şi îşi găsesc aplicații în diferite domenii ale ştiinței şi ingineriei. Prezentăm pe scurt în continuare tipurile de rețele Petri folosite în această lucrare[94].

4.3.1. Rețele Petri autonome[95].

Acestea sunt rețelele Petri clasice numite adeseori simplu rețele Petri şi sunt definite de (4.1) şi (4.2). Pozițiile şi tranzițiile acestor rețele nu sunt interpretate, deci nu există intrări şi ieşiri. Sistemul cu evenimente discrete modelat de o rețea Petri autonomă este caracterizat numai de variabilele sale de stare exprimate prin vectorul de marcaj M. Tranzițiile aceste rețele se declanşează secvențial şi instantaneu de fiecare dată când sunt validate. Nu există două tranziții declanşate simultan (sau în paralel). Mărcile pot rămâne în poziții orice durată de timp. Acest tip de rețea este utilizat numai pentru studiul proprietăților structurale, care nu depind de timp, ale sistemelor cu evenimente discrete şi ale sistemelor discrete logice. Trecerea de la un marcaj la altul se face pas cu pas, declanşând secvențial tranzițiile.

4.3.2. Rețele Petri interpretate.

Aceste rețele au fost prezentate în paragraful 4.1. Tranzițiile şi pozițiile sunt interpretate prin ataşarea unor evenimente şi acțiuni care reprezintă semnalele logice de intrare şi ieşire ale sistemului. Ori de câte ori o poziție este activă acțiunile ataşate ei se execută. Evenimentul logic ataşat tranziției o interpretează în sensul că tranziția se declanşează numai dacă evenimentul a apărut, adică ia valoarea logică 1. Evenimentul care interpretează tranziția poate fi o funcție logică de alte evenimente externe sistemului sau poate fi un eveniment particular, numit evenimentul prezent întotdeauna şi notat cu e. Rețelele Petri interpretate sigure modelează un tip particular de sisteme cu evenimente discrete numite sisteme discrete logice.

4.3.3. Rețele Petri temporizate.

Temporizarea unei tranziții se face prin interpretarea ei cu un eveniment special care apare după un interval determinat de timp de la validarea ei. O rețea Petri temporizată are toate tranzițiile temporizate. Dacă temporizarea tuturor tranzițiilor se face cu un interval de timp de durată zero se obține o rețea Petri autonomă. În *Fig. 4.6* este prezentat un exemplu de rețea Petri temporizată.

---

[94] Alte tipuri de rețele Petri sunt rețelele Petri stochastice, rețelele Petri colorate şi altele.

[95] Se mai numeşte rețea Petri netemporizată sau rețea Petri poziții – tranziții.

4.4. Analiza structurală a sistemelor cu evenimente discrete.

Un rol foarte important în comportarea unui sistem cu evenimente discrete îl au proprietățile lui structurale puse în evidență de rețeaua Petri care îl modelează. Aceste proprietăți depind de N din (4.2).

4.4.1. Structuri tip folosite la modelarea cu rețele Petri

Cele mai importante structuri care apar într-o rețea Petri sunt: secvența (*Fig. 4.10*a), alegerea *(Fig. 4.11* a), saltul (*Fig. 4.10* b), repetarea (*Fig. 4.10* c), convergența (*Fig. 4.12*a), paralelismul (*Fig. 4.13* a) şi sincronizarea (*Fig. 4.14* a).

Secvența (*Fig. 4.10*a) reprezintă o succesiune liniară de poziții care sunt activate una după alta prin declanşarea tranzițiilor.

Alegerea (*Fig. 4.11* a) permite trecerea de la o poziție activă în amonte la una dintre alte poziții active situate în aval. Existența unei structuri tip alegere implică întotdeauna un conflict structural sau efectiv, studiat în 4.4.7, care poate fi eliminat prin interblocare 4.4.8

Saltul (*Fig. 4.10* b)are loc într-o secvență liniară atunci când se doreşte evitarea activării unei poziții. Saltul implică şi o alegere, astfel încât trebuiesc rezolvate toate problemele legate de conflictul tranzițiilor care apare la această structură.

Repetarea (*Fig. 4.10* c) este şi ea o alegere particulară care permite reactivarea unei poziții dintr-o secvență. Această structură conduce la o rețea Petri impură (degenerată) şi trebuie transformată sau evitată aşa cum se arată în 4.4.3.

Convergența (*Fig. 4.12* a) este structura duală alegerii şi este folosită atunci când se doreşte ca două sau mai multe acțiuni să se termine în aceiaşi manieră.

Paralelismul structural[96] (*Fig. 4.13* a), contrar alegerii care stabileşte o singură acțiune care se poate realiza, permite mai multor activități independente să se deruleze în paralel.

Sincronizarea (Fig. 4.14 a) permite aşteptarea sfârşitului mai multor acțiuni care se realizează în paralel pentru continuarea lor printr-o aceiaşi altă acțiune

4.4.2. Rețele Petri ordinare.

O rețea Petri se numeşte ordinară dacă ponderea tuturor arcelor sale este egală cu unu. Matricele Pre şi Post au toate elementele egale cu 1. Un exemplu se prezintă în *Fig. 4.5*. Se observă că ponderea unitară a arcelor nu mai este specificată.

[96] Există la rețeaua Petri şi un paralelism tip conveier studiat în 4.7. Grafcetul, o rețea Petri Particulară, permite un paralelism realizat cu ajutorul structurii tip alegere 4.9.6.

4.4.3. Rețele Petri pure.

Rețeaua Petri este *pură* sau *nedegenerată* dacă nu are *bucle elementare* formate dintr-o singură poziție şi o singură tranziție. In acest caz se asigură independența între consumarea şi producerea mărcilor (resurselor) în momentul declanşării tranzițiilor. Bucla din *Fig. 4.5* este formată din poziția P1 şi tranziția T4. Condiția de puritate este:

$$\forall p,t: \qquad \Pr e(p,t).Post(p,t) = 0 \tag{4.12}$$

Buclele elementare care impurifică rețeaua sunt, în general, inutile pentru modelarea sistemelor discrete logice. Multe dintre rezultatele teoretice obținute în cadrul analizei rețelelor Petri sunt valabile numai pentru rețelele pure. Din această cauză este recomandabil ca buclele elementare să fie eliminate sau transformate. Se observă că o buclă elementară introduce şi un conflict structural 4.4.7. Eliminarea acestuia se poate face prin interblocare, ca în paragraful 4.4.8, iar bucla elementară devine de prisos putând fi înlăturată. Transformarea buclei elementare se realizează introducând suplimentar în buclă o poziție si o tranziție fictive. In felul acesta rețeaua Petri nu mai este degenerată.

4.4.4. Poziția sursă sau receptor.

O poziție fără nici o tranziție de intrare se numeşte *poziție sursă*, de exemplu P_1 în Fig. 4.1. Poziția fără nici o tranziție de ieşire este o *poziție receptor*.

4.4.5. Tranziția validată[97]*.*

Condiția de *validare* a unei tranziții t_1, care asigură existența resurselor ce urmează să fie consumate la declanşarea tranziției, se exprimă astfel:

$$t_1 \in T, \quad \forall p \in P: \qquad M(p) \geq \Pr e(p,t_1) \tag{4.13}$$

Cu alte cuvinte, numărul mărcilor existente în pozițiile precedente tranziției este suficient pentru a fi consumate la declanşarea sa.

De exemplu, pentru rețeaua Petri din *Fig. 4.2*

$$M_0(p) \geq \Pr e(p,T_1) \Rightarrow \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \geq \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \tag{4.14}$$

şi deci tranziția T_1 este validată pentru marcajul inițial M_0.

[97] Enabled (eng.)

4.4.6. Tranziția declanşată[98].

O tranziție este declanşată dacă este validată şi evenimentul care o interpretează a apărut (are valoarea logică unu). Prin declanşare se îndepărtează un număr de mărci din fiecare poziție de intrare şi se adaugă un număr de mărci în fiecare poziție de ieşire. Numărul mărcilor îndepărtate, respectiv adăugate, este determinat de ponderea arcelor de intrare, respectiv ieşire, din tranziție.

4.4.7. Conflictul structural şi conflictul efectiv al tranzițiilor.

Două tranziții sunt în *conflict* structural dacă au cel puțin o poziție în comun la intrare. Aceleaşi tranziții sunt în plus şi în conflict în *conflict efectiv* pentru un marcaj M dacă amândouă sunt declanşabile şi resursele sunt insuficiente. Tranzițiile T2 şi T4 din *Fig. 4.5* sunt în conflict structural şi efectiv. Condiția de conflict structural a tranzițiilor oarecare t_1 şi t_2 este:

$$\exists p: \qquad \Pr e(p,t_1).\Pr e(p,t_2) \neq 0 \tag{4.15}$$

Tranzițiile T_2 şi T_4 din *Fig. 4.5*sunt în conflict structural deoarece:

$$P_2: \qquad \Pr e(P_2,T_2).\Pr e(P_2,T_4) = 1 \tag{4.16}$$

Aceleaşi două tranziții t_1 şi t_2 sunt în plus şi în conflict efectiv pentru un marcaj M(p) dacă amândouă sunt declanşabile şi resursele sunt insuficiente:

$$M(p) < \Pr e(p,t_1) + \Pr e(p,t_2) \tag{4.17}$$

Dacă poziția P_2 este activă fiecare dintre tranzițiile T_2 şi T_4 este validată şi deci declanşabilă. Condiția (4.17) este verificată pentru poziția B:

$$M(P_2) < \Pr e(P_2,T_2) + \Pr e(P_2,T_4) = 2 \tag{4.18}$$

şi deci tranzițiile T_2 şi T_4 sunt în conflict efectiv. Aceasta înseamnă că dacă motorul este oprit în P_2 şi se apasă simultan butoanele de pornire p ăi oprire o, amândouă tranzițiile T_2 şi T_4 sunt declanşate ducând la două situații complect diferite şi incompatibile: în P_3 motorul este pornit iar în P_2 motorul este oprit. Deoarece regulile de funcționare ale rețelei Petri interpretate impun ca la un anumit moment de timp să poată fi declanşată numai o singură tranziție, rezultă că în cazul studiat se va declanşa T_2 sau T_4 în funcție de factori aleatori care fac ca unul dintre evenimentele care le interpretează să apară cu o fracțiune de secundă mai devreme.

[98] Fired (eng).

Distincția dintre conflictele structurale şi efective are sens numai pentru rețelele Petri interpretate sau temporizate. În cazul unei rețele Petri autonome existența unui conflict structural conduce la o comportare nedeterministă a sistemului pe care îl modelează.

4.4.8. Interblocarea prin interpretare.

Eliminarea conflictului efectiv între două tranziții se poate face condiționând astfel evenimentele care le interpretează încât să nu poată apărea simultan. Această acțiune se numeşte *interblocare.* Pentru rețeaua din *Fig. 4.5* s-a impus condiția ca tranziția T_2 să fie interpretată de un eveniment care apare, sau ia valoarea logică unu, numai atunci când se apasă pe butonul de pornire p şi *nu* se apasă butonul de oprire o.

$$T_2 \leftarrow p.\bar{o} \tag{4.19}$$

Prin interblocare se pot înlătura şi buclele care fac rețeaua degenerată (impură). De exemplu, interpretarea tranziției T_2 din (4.19) face inutila prezența buclei care începe şi se termină în P_2.

4.5. Analiza comportamentală a sistemelor cu evenimente discrete.

4.5.1. Ecuația de stare.

Declanşarea diferitor tranziții[99] a rețelei Petri interpretate face ca starea sistemului cu evenimente discrete SED pe care îl modelează, descrisă de marcajul său M(p), să se modifice. De exemplu, în *Fig. 4.2* starea este dată de M_0 definit de (4.7). Declanşarea tranziției T_1 conduce la starea din *Fig. 4.3* cu marcajul M_1 din (4.20). Asemănător declanşarea lui T_2 provoacă apariția marcajului M_2 cu valorile date de (4.20). Se poate spune că o secvență de declanşări de tranziții *s* definită de (4.21) conduce sistemul cu evenimente discrete din starea M_0 în starea *M*.

$$T_1 \Rightarrow \quad M_1(p) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \qquad T_2 \Rightarrow \quad M_2(p) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.20}$$

Pentru secvența *s* (4.21) se defineşte *vectorul caracteristic* s^* corespunzător (4.22) definit ca vectorul[100] cu indici în *t,* definit de (4.22), şi componente care reprezintă numărul de apariții a fiecărei tranziții în secvența *s.*

$$s = T_1 T_2 T_3 T_2 \tag{4.21}$$

t	**s^*(t)**
T_1	*1*
T_2	*2*
T_3	*1*

declanşează dacă este validată şi evenimentul care o interpretează a apărut.

Acesta mai este numit vectorul de numărare a execuțiilor.

$$s^* = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \tag{4.22}$$

Legătura dintre două marcaje a unei rețele Petri interpretate *pure* este dată de *ecuația fundamentală de stare*:

$$M = M_0 + C.s^* \tag{4.23}$$

în care C este matricea de incidență definită de (4.8).

Această ecuație permite calculul *stării* unui sistem cu evenimente discrete (marcajul M) dacă se cunoaşte starea inițială M_0 a sistemului, *funcția de tranziție* a stărilor (matricea de incidență C) şi *intrarea* sub forma unei secvențe de tranziții *declanşabile s* determinată de *vectorul caracteristic* s^*. Folosind ecuația fundamentală sistemele cu evenimente discrete pot fi analizate cu ajutorul metodelor algebrei liniare, în mod asemănător cu analiza sistemelor continue.

Pentru exemplul considerat până acum referitor la rețeaua din *Fig. 4.2* cu matricea de incidență (4.11) şi secvența (4.21) rezultă noua stare:

$$M = M_0 + C.s^* = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -2 & 0 & 0 \\ 1 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -2 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.24}$$

Trebuie subliniat că ecuația fundamentală (4.23) se referă numai la rețelele Petri interpretate *pure* şi este valabilă numai pentru secvențe de tranziții *declanşabile*, adică de tranziții care îndeplinesc, fiecare separat, condiția de *validare* (4.8). Matricea de incidență C depinde numai de structura N a rețelei definită de (4.2).

4.5.2. Graful marcajelor accesibile.

Graful marcajelor accesibile al unei rețele Petri RP=(N,M_0) este graful care are drept vârfuri mulțimea marcajelor M_i care se obțin plecând din M_0 şi drept arce tranzițiile corespunzătoare.

4.6. Performanțele sistemelor cu evenimente discrete.

Performanțele SED sunt determinate de anumite valori pe care o iau mărimile de ieşire. Aceste mărimi sunt la rândul lor sunt determinate de starea SED, adică de marcajul său M.

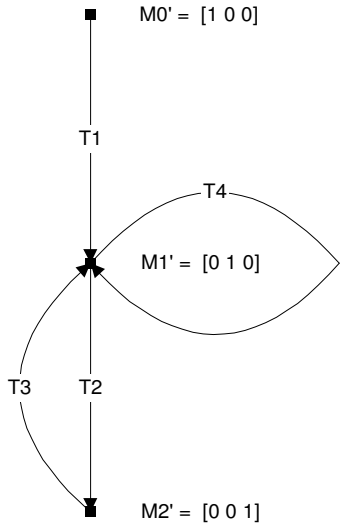


Fig. 4.4 Graful marcajelor accesibile M' pentru reţeaua Petri din Fig. 4.5

4.6.1. Reversibilitatea[101]

Un interes deosebit, din punct de vedere practic, îl prezintă secvenţele de intrare *s* pentru care starea finală a sistemului cu evenimente discrete este egală cu starea sa iniţială, adică:

$$M = M_0 \qquad (4.25)$$

Din relaţiile (4.23) şi (4.25) rezultă condiţia de *reversibilitate staţionară* a unei secvenţe s^*

$$C.s^* = 0 \qquad (4.26)$$

Tranziţiile care formează secvenţa care îndeplineşte condiţia (4.26) formează o componentă reversibilă staţionar a reţelei Petri interpretate. Dacă această componentă cuprinde *toate* tranziţiile reţelei Petri staţionare atunci reţeaua se numeşte reversibilă staţionar. Sistemele cu evenimente discrete *automate* trebuie să fie caracterizate de reţele Petri reversibile staţionar. In felul acesta se asigură repetabilitatea anumitor activităţi.

[101] Reversibility (eng.)

Dacă se aplică relația (4.26) la rețeaua din *Fig. 4.2* se obține:

$$Cs^* = \begin{bmatrix} -2 & 0 & 0 \\ 1 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} . \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = 0 \qquad (4.27)$$

Soluția acestui sistem de ecuații va fi $s_1=0$ şi $s_2=s_3$. Deci un vector caracteristic care satisface condiția (4.26) este:

$$s^* = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \qquad (4.28)$$

Componenta reversibilă a rețelei va fi compusă din tranzițiile T_2 şi T_3 după cum rezultă şi din *Fig. 4.2*.

4.6.2. Mărginirea şi siguranța.

O rețea Petri este *mărginită*[102] dacă numărul de mărci din fiecare poziție are o valoare mărginită pentru toate situațiile de funcționare. Vectorul caracteristic s^* al secvenței de intrare s definit de (4.22) este un model matematic care condensează şi reduce informația referitoare la intrarea sistemului cu evenimente discrete. Un pas mai departe în această direcție constă în efectuarea produsului scalar între vectorul marcaj şi un vector f care ponderează numărul de mărci din fiecare poziție a rețelei. Ecuația fundamentală (4.23) va deveni atunci:

[102] Bounded (eng.)

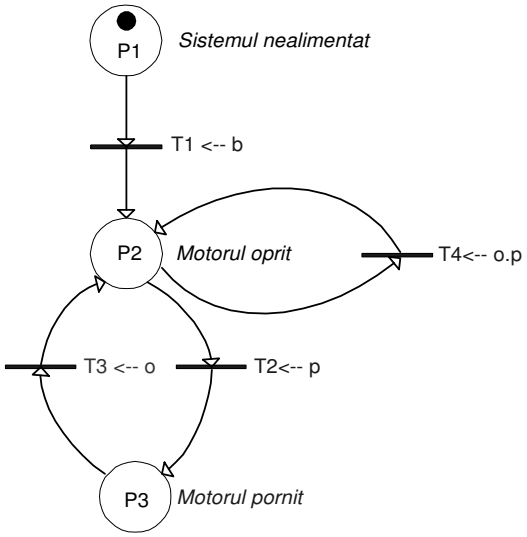


Fig. 4.5 O rețea Petri interpretată ordinară, sigură, degenerată, neviabilă, cu conflict structural şi efectiv între T_2 şi T_4 .

$$f^T.M = f^T.M_0 + f^T.C.s^* \tag{4.29}$$

în care f^T este transpusul vectorului f.

Se demonstrează că rețeaua este mărginită dacă există un vector *f* pozitiv astfel încât să fie îndeplinită condiția:

$$\exists f > 0, \quad f^T C \le 0 \tag{4.30}$$

Dacă se aplică această relație la rețeaua din *Fig. 4.2* se obține:

$$\begin{aligned} f^T C &= \begin{bmatrix} f_A & f_B & f_C \end{bmatrix} \cdot \begin{bmatrix} -2 & 0 & 0 \\ 1 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} = \\ &\begin{bmatrix} (-2f_A + f_B) & (-f_B + f_C) & (f_B - f_C) \end{bmatrix} = 0 \end{aligned} \tag{4.31}$$

Soluția acestui sistem de ecuații liniare este:

$$2f_A = f_B = f_C \qquad (4.32)$$

Deci există un vector *f* pozitiv şi rețeaua este mărginită.

Rețeaua Petri este denumită *sigură*[103] sau *binară sau logică* pentru un anumit marcaj M_0 dacă toate pozițiile au un număr de mărci egal sau mai mic decât unu pentru toate situațiile de funcționare apărute plecând de la acest marcaj inițial. In *Fig. 4.5* se prezintă o rețea Petri interpretată sigură. Deoarece pozițiile pot avea maximum o marcă, arcele orientate nu mai sunt evaluate, aşa cum se întâmplă în *Fig. 4.5*. Rețeaua Petri este în acest caz ordinară. Rețelele Petri sigure modelează numai operațiile din sistem nu şi distribuirea resurselor.

Performanța de mărginire este asemănătoare cu performanța de stabilitate a sistemelor continue. În ambele cazuri ieşirea sistemului trebuie să fie finită.

4.6.3. Viabilitatea.

O rețea Petri RP=(N,M_0) este *viabilă*[104] dacă pentru orice marcaj M realizat pornind din M_0 se poate declanşa în continuare orice tranziție a rețelei N, eventual după declanşarea unui număr finit de alte tranziții. Dacă se ajunge la un marcaj pentru care nici o tranziție a rețelei nu mai poate fi executată, acesta se numeşte *blocaj*[105].

Performanțele de reversibilitate, mărginire şi viabilitate joacă un rol important în studiul rețelelor Petri. Aceste performanțe nu depind una de alta.

[103] Safe (eng.)

[104] Live (eng.)

[105] Deadlock (eng.)

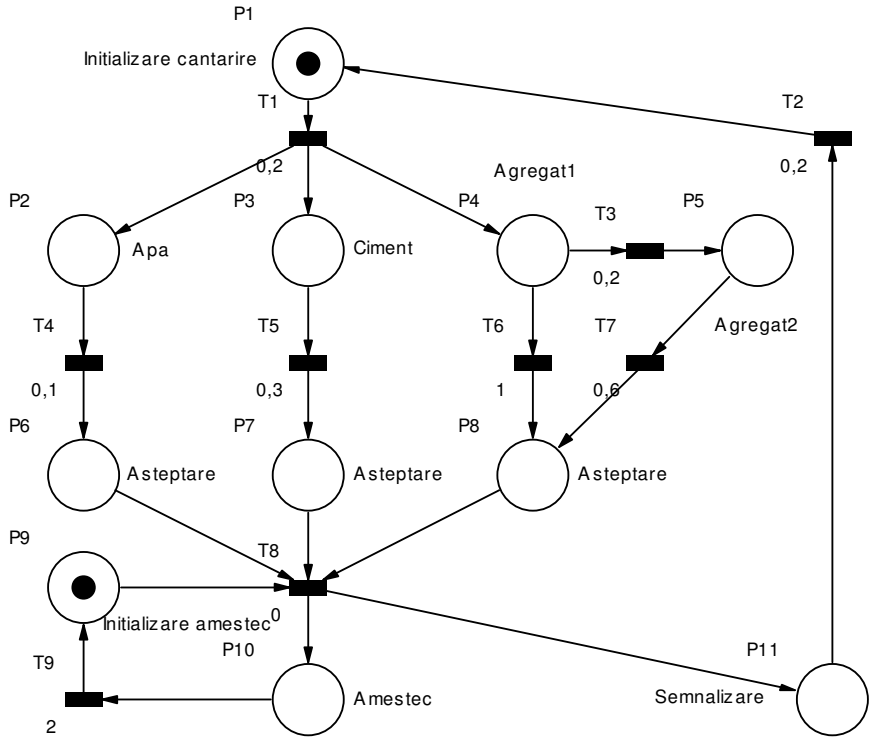


Fig. 4.6 Modelul automatului unei staţii de betoane sub forma unei reţele Petri temporizate şi sigure.

4.7. Sisteme cu evenimente discrete şi evoluţie paralelă.

Un exemplu de sistem discret logic va permite explicarea conceptului de paralelism.

Automatul care conduce o staţie de betoane este modelat cu ajutorul unei reţele Petri temporizate sigure în *Fig. 4.6*. El trebuie să asigure cântărirea apei, cimentului, agregatelor şi să comande amestecul lor.

Fiecare poziţie şi tranziţie a reţelei Petri are un nume şi este interpretată. Toate tranziţiile au ataşate evenimente[106] sub forma unor temporizări. De exemplu, evenimentul tranziţiei T1, specificat în dreptul ei, apare după $\tau = 0{,}2$ minute de la validarea[107] sa. Reţelele Petri de acest tip se numesc *temporizate.* Poziţiile P1 şi P9 sunt active în momentul iniţial şi atunci când cântarele şi malaxorul sunt goale. Să vedem dacă automatul are performanţa de reversibilitate, adică revine în starea iniţială după o secvenţă finită de operaţiuni. În situaţia din *Fig. 4.6* este validată numai tranziţia T1 care se declanşează atunci când apare evenimentul de temporizare care o interpretează, adică după 0,2 minute. După declanşarea sa poziţia P1 devine inactivă iar poziţiile P2, P3, P4 sunt active şi se execută acţiunile care le

[106] Sunt interpretate într-un mod specific.

[107] O tranziţie este validată dacă toate poziţiile precedente sunt active.

interpretează, adică se umplu recipientele de apă, ciment şi agregat1. Cântărirea se face prin temporizare, adică, de exemplu, cantitatea dorită de apă se obține după o umplere a rezervorului de 0,1 minute. Asemănător, pentru un debit constant de umplere, cantitatea dorită de ciment se obține după 0,3 minute.

În cazul poziției P4 avem un conflict structural deoarece atunci când este activă sunt validate tranzițiile T6 şi T3. Conflictul nu este efectiv însă deoarece evenimentele care interpretează tranzițiile nu apar simultan, având temporizări diferite de 1 şi 0,2 minute. Deci, după activarea simultană a pozițiilor P2,P3 şi P4 devin active pe rând P6, P5, P7 şi P8. În P5 se cântăreşte agregatul cu numărul 2 iar în restul pozițiilor se aşteaptă terminarea cântăririi tuturor componentelor. Cântărirea agregatelor se face prin acumulare în acelaşi recipient.

Se observă că operațiile de cântărire a apei, cimentului şi agregatelor se execută în paralel, adică în acelaşi timp[108]. Acest tip de execuție simultană se numeşte *paralelism general.*

La terminarea cântăririi componentelor se validează tranziția T8, deoarece P6, P7, P8 şi P9 sunt active şi se declanşează imediat deoarece este interpretată cu o temporizare zero. Pozițiile P10 şi P11 devin active. Acțiunile care le interpretează specifică începerea amestecului componentelor betonului şi semnalizarea terminării cântăririi. Tranzițiile T9 şi T2 sunt validate de se vor declanşa la intervale de timp diferite, respectiv 2 şi 0,2 minute. Deci mai întâi se declanşează P1 care inițializează cântărirea pentru şarja a doua de betoane.

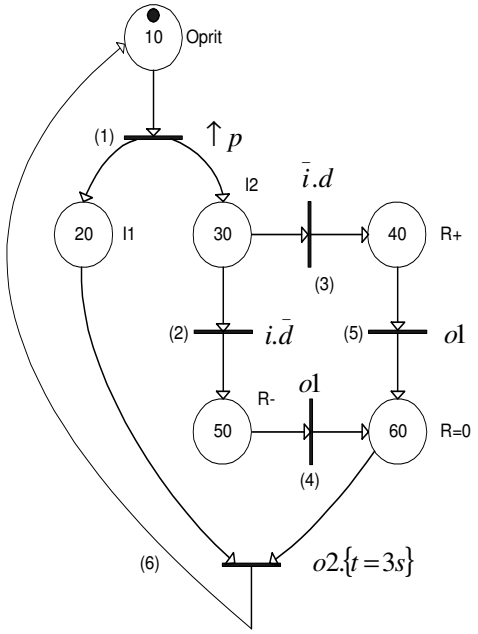
Apa, cimentul şi agregatele şarjei a doua se transportă în recipientele respective şi se cântăresc prin temporizare în timp ce componentele primei şarje se amestecă. Deci cântărire pentru şarja a doua şi amestecul pentru prima şarjă se execută în acelaşi timp, în paralel. Această execuție simultană se numeşte *paralelism conveier*[109].

A doua cântărire se termină şi amestecul primei şarje nu este încă terminat. Se aşteaptă în pozițiile P6, P7 şi P8 ca să se inițializeze un nou amestec în poziția P9. Când aceasta se întâmplă se reiau operațiile simultane de cântărire şi amestec.

[108] Cântărirea agregatelor se face în paralel cu a apei şi cimentului, dar cele două tipuri de agregate se cântăresc secvențial prin acumulare.

[109] Conveyer (eng.)

a)



b)

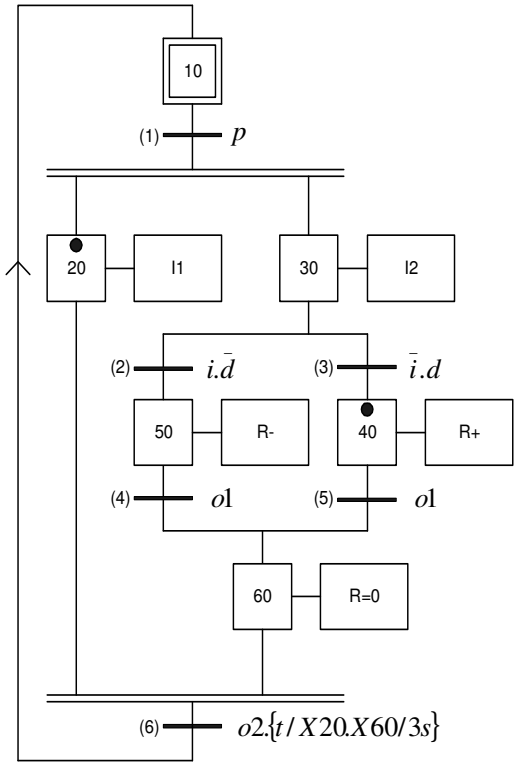


Fig. 4.7 Modelul unui sistem discret logic, format din instalațiile I1, I2 şi robinetul R, prezentat sub forma unei rețele Petri interpretate, sigure, viabile şi fără conflicte a) şi sub forma unui grafcet b).

Marcajul inițial format din pozițiile P1 şi P9 active nu se mai obține niciodată. Rețeaua Petri din *Fig. 4.6* nu este *reversibilă.* Pentru evitarea acestui fenomen la terminarea cântăririi numărului dorit de şarje se modifică evenimentul care interpretează tranziția T1 astfel încât aceasta să nu se mai declanşeze automat, cu o temporizare de 0,1 minute, după activarea poziției P1.

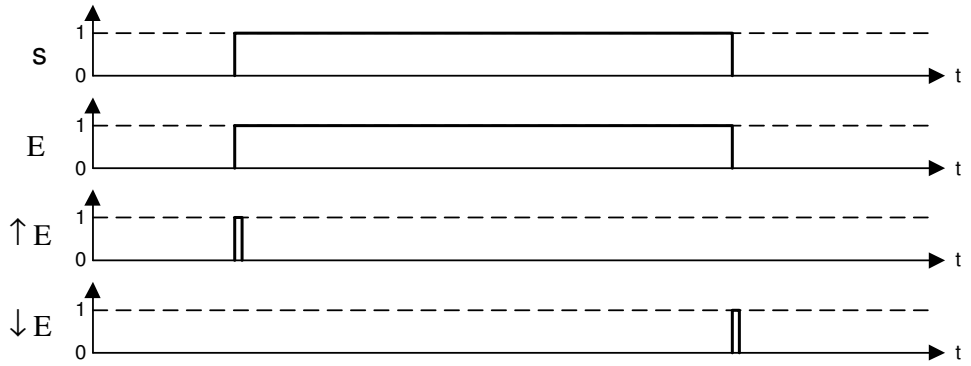


Fig. 4.8 Semnalul logic s de intrare în sistem şi evenimentele tip nivel E şi tip impuls ↑E, ↓E ataşate.

4.8. Maşina de stare.

Maşina de stare[110] este un sistem cu evenimente discrete care poate fi modelat printr-o rețea Petri la care fiecare tranziție este legată la exact o poziție de intrare şi o poziție de ieşire prin arce cu ponderea 1.

$$\forall t \in T \qquad \sum_{p \in P} \Pr e(p,t) = 1 \qquad \sum_{p \in P} Post(p,t) = 1 \tag{4.33}$$

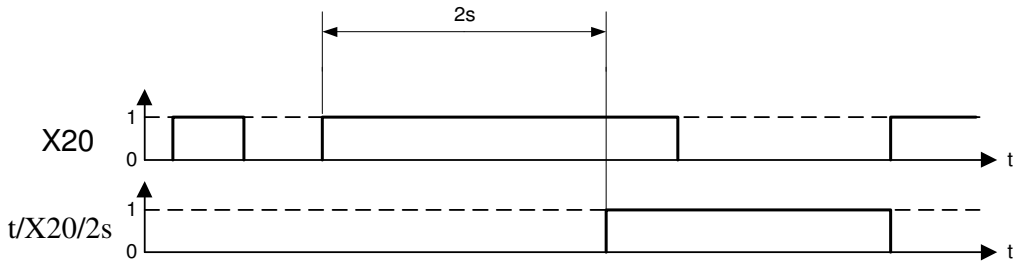


Fig. 4.9 Variabila t/X20/2s de temporizare față de activarea etapei 20

[110] State machine (eng.)

Dacă maşina de stare este sigură, atunci marcajul iniţial M_0 conţine o singură marcă. Aceasta se va deplasa în cadrul evoluţiei sistemului din poziţie în poziţie indicând starea activă curenta. O maşină de stare interpretată şi sigură mai poartă numele de *automat finit*. In acest caz graful marcajelor accesibile este izomorf cu graful reţelei obţinut prin ştergerea tranziţiilor. Reciproc, dacă graful marcajelor accesibile ale unei reţele Petri oarecare este finit i se poate asocia o maşină de stare. Deoarece maşina de stare poate fi implementată relativ uşor rezultă că orice reţea Petri cu graful marcajelor accesibile *finit* poate fi implementată.

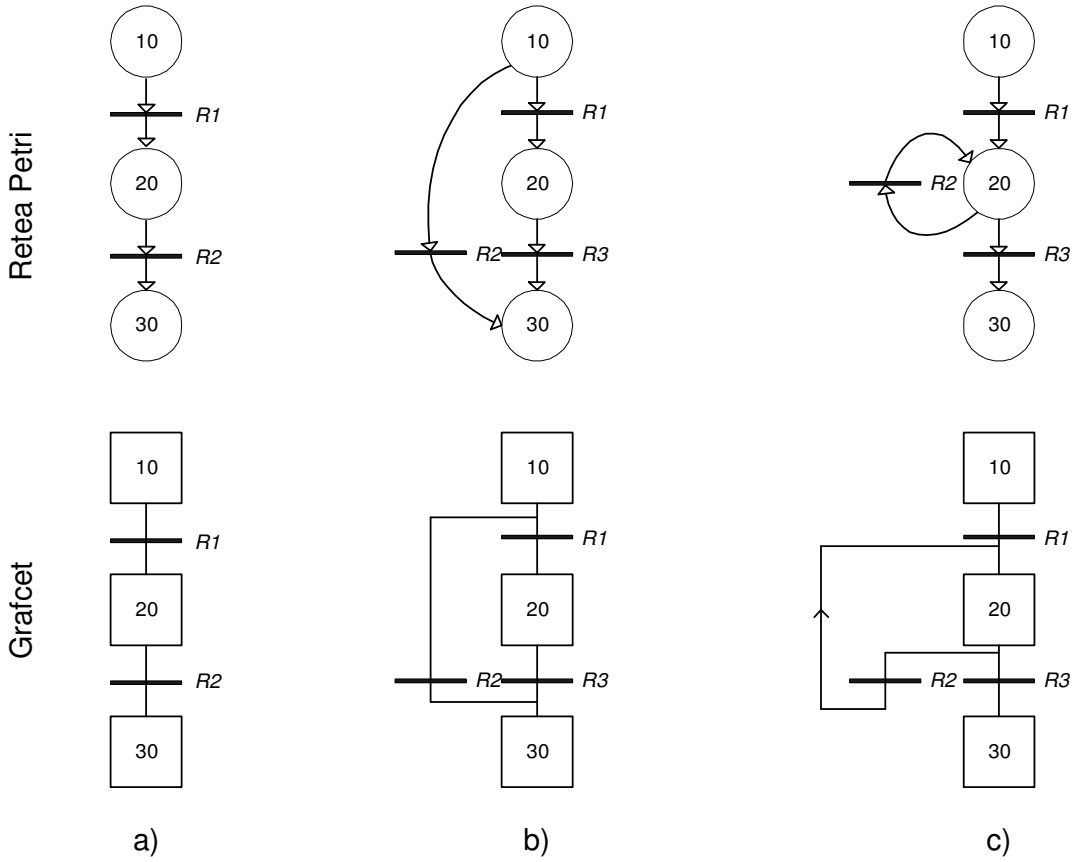


Fig. 4.10 Structurile tip secvenţă a), salt b) şi repetare c) la reţeaua Petri şi grafcet.

Condiţia de *viabilitate* a maşinii de stare este dată de teorema lui Commoner: Condiţia necesară şi suficientă pentru ca reţeaua Petri să fie viabilă este ca graful obţinut din reţea prin ştergerea tranziţiilor să fie tare conex şi marcajul iniţial să conţină cel *puţin* o marcă.

Condiţia de *siguranţă* pentru o maşină de stare impune ca marcajul iniţial M_0 să aibă cel *mult* o marcă.

O maşină de stare *viabilă* este şi *sigură* dacă şi numai dacă marcajul iniţial M_0 are o *singură* marcă.

4.9. Grafcetul.

4.9.1. Trecerea de la rețeaua Petri la grafcet.

Rețelele Petri interpretate, sigure, viabile şi fără conflicte efective sunt numite *conforme* şi coincid cu o clasă de grafceturi frecvent utilizate în practică. Cuvântul *grafcet*[111] provine din inițialele GRAF de Comandă Etape - Tranziții – *GRAFCET*[112].

Grafcetul este un model folosit foarte mult pentru sistemele discrete logice, a fost standardizat internațional[113] şi a devenit unul din instrumentele de bază pentru programarea automatelor programabile logice. Trecerea de la o rețea Petri conformă la un grafcet se face imediat, după cum se vede în *Fig. 4.7*, deosebirile constând numai în formalismul grafic[114].

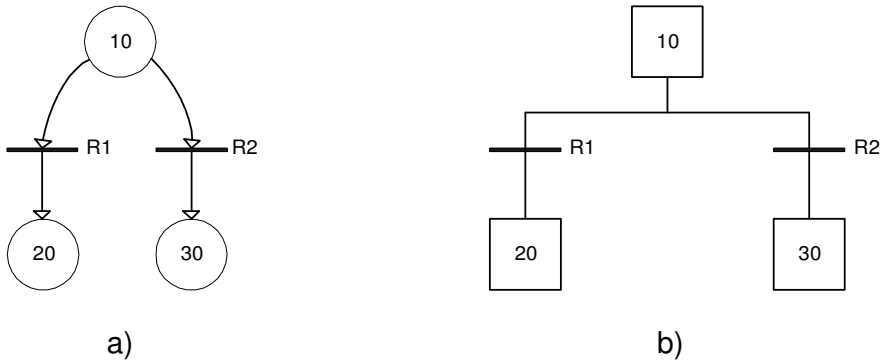


Fig. 4.11 Structura tip alegere la rețeaua Petri a) şi grafcet b).

Exemplul din *Fig. 4.7* se referă la modelul unui sistem discret logic format din două instalații I1, I2 şi un robinet R. La acționarea butonului de pornire *p* ambele instalații pornesc simultan în paralel După aceasta se închide sau se deschide, mai mult sau mai puțin, cu ajutorul unui servomotor reversibil, robinetul R. Pentru aceasta se acționează butonul de închidere *i* sau butonul de deschidere *d.* Oprirea închiderii sau deschiderii robinetului se face cu butonul *o1*. Oprirea tuturor instalațiilor se face cu butonul *o2* şi sistemul ajunge în starea inițială modelată de poziția 10 la rețeaua Petri sau etapa[115] 10 la grafcet. Se observă că pozițiile sunt reprezentate prin cercuri iar etapele grafcet prin pătrate.

Tranziția (1) realizează o *distribuție ŞI* iar tranziția (6) realizează o *joncțiune ŞI*. Din etapa 30 urmează o *distribuție SAU* iar etapa 60 realizează o *joncțiune SAU*. Se observă în

---

[111] Prin proveniența sa *grafcet* este un substantiv românesc, deşi coincide cu cuvântul francez. Din această cauză, la fel ca în limba franceză, el se *pronunță* şi se *foloseşte* după regulile limbii române.

[112] Grafcet – Graphe de Commande Etape – Transition (fr.).

SFC – Sequential Function Chart (eng.).

[113] IEC 848 (Function Chart for Control System – FCCS), 1987.

[114] Trecere inversă, de la grafcet la o rețea Petri, nu este directă deoarece grafcetul posedă facilități suplimentare de modelare, cum ar fi macro-etapa, care nu se regăsesc la rețeaua Petri.

[115] Step (eng.)

Fig. 4.7 modul diferit în care rețeaua Petri şi grafcetul prezintă distribuțiile şi joncțiunile tip SI şi SAU.

Tranzițiile (2) şi (3) sunt în conflict structural pentru rețeaua Petri. Conflictul nu este însă efectiv deoarece s-a realizat o interblocare între acționarea butoanelor *i* şi *d*. Noțiune de conflict structural şi efectiv nu există la grafcet. Dacă n-ar fi existat interblocarea, etapele 40 şi 50 ar fi putut fi activate simultan.

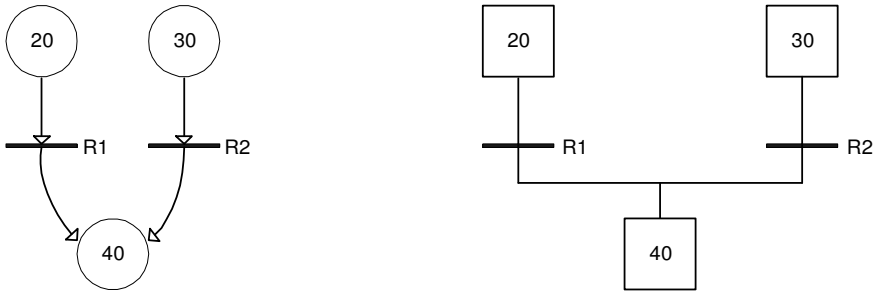


Fig. 4.12 Structura tip convergență la rețeaua Petri a) şi grafcet b).

4.9.2. Etape, tranziții şi legături orientate.

Etapa din grafcet corespunde poziției (locației) din rețeaua Petri şi modelează o situație în care comportamentul sistemului modelat, sau numai a unei părți ale sale, rămâne neschimbat în raport cu intrările şi ieşirile sale.

Etapa inițială, activă la momentul de timp inițial, este specificată la grafcet printr-un pătrat dublu. Pentru grafcetul din *Fig. 4.7* b) etapa inițială are numărul 10 iar etapele active la momentul de timp curent sunt specificate prin plasarea unei mărci în pătratele corespunzătoare cu numărul 20 şi 40. Aceasta înseamnă că grafcetul din figură prezintă situația sistemului discret logic în care funcționează instalațiile I1, I2 şi robinetul R se deschide. La rețeaua Petri nu se specifică starea inițială iar pentru cazul din *Fig. 4.7* a) sistemul modelat se găseşte în starea de la momentul de timp curent specificată prin plasarea unei mărci în poziția corespunzătore cu numărul 10.

Acțiunile care interpretează etapele grafcetului sunt trecut în dreptunghiuri ataşate de acestea. Pentru etapele inițiale se consideră că acțiunile implicite sunt cele care opresc toate echipamentele şi se specifică, dacă este cazul, numai acțiunile diferite de acestea.

Fiecărei etape i se asociază o variabilă de stare logică notată X_i, în care i este numărul etapei. Această variabilă ia valoarea logică 1 dacă etapa este activă sau 0 dacă etapa este inactivă.

Tranzițiile şi evenimentele care le interpretează sunt reprezentate la fel în cazul rețelei Petri şi a grafcetului. Denumirea tranzițiilor este formată de un număr plasat între paranteze rotunde. De exemplu, tranziția (5) este interpretată de evenimentul *o1,* o variabilă logică care ia valoarea logică 1 (adevărat) dacă se acționează butonul de oprire cu numărul 1. La grafcet

tranzițiile sunt formate din linii *orizontale* îngroşate, pe când rețelele Petri reprezintă tranzițiile prin dreptunghiuri[116] înnegrite care pot fi situate oricum.

Arcele orientate sunt formate la grafcet numai din *linii drepte şi verticale* cu sensul de sus în jos. Din această cauză săgețile nu mai sunt prezente. Excepție fac doar arcele orientate de întoarcere la starea inițială. La rețeaua Petri, *Fig. 4.7* a), arcele orientate pot avea orice formă, pot fi orientate oricum şi sensul este indicat prin săgeți.

4.9.3. Interpretarea tranzițiilor.

La grafcet fiecărei tranziții îi este asociată o receptivitate R_i, în care i este numărul tranziției. Receptivitatea este o funcție logică de evenimente, stări ale etapelor şi temporizări.

Evenimentele sunt, ca şi la rețeaua Petri, variabile logice ataşate semnalelor de intrare. In *Fig. 4.8* se observă că pentru un semnal logic s(t) de intrare se pot defini variabilele logice tip nivel E şi tip impuls ↑E, ↓E care apar pe frontul crescător, respectiv descrescător al semnalului E.

Variabilele de stare logică ataşate fiecărei etape sunt notate cu X_i, în care i este numărul etapei. Aceste variabile iau valoarea logică 1 atâta timp cât etapa este activă.

Variabilele de temporizare se definesc față de momentul în care o etapă devine activă. In Fig. 4.9 se prezintă variabila de temporizare t/X20/2s care are valoarea logică 1 după o întârziere de 2 secunde de la activarea etapei cu numărul 20. Se observă că temporizarea apare numai dacă etapa este activă mai mult de 2 secunde. In momentul în care etapa se reactivează temporizarea revine la valoarea logică 0.

4.9.4. Interpretarea etapelor.

Fiecărei etape i se poate ataşa o variabilă logică funcție de starea etapei la care este ataşată, de semnale de intrare şi de o temporizare. De exemplu

$$A20 = X20.a.t \,/\, X20/5s \qquad (4.34)$$

De exemplu, acțiunea A20 ataşată etapei X20 şi definită de (4.34) apare, adică ia valoarea logică 1 dacă etapa cu numărul 20 este activă, adică X20=1, semnalul de intrare a=1 şi temporizarea *t/X20/5s* a fost realizată.

---

[116] La rețelele Petri generale tranzițiile consumă un număr resurse (mărci) din pozițiile precedente şi produc un alt număr de mărci în pozițiile posterioare. Deci ele au consistență şi apar ca nişte dreptunghiuri ca în *Fig. 4.6*. Pentru rețelele Petri sigure, cu maximum o marcă în fiecare poziție, folosite la modelarea sistemelor discrete logice, se adoptă, în general, o reprezentare apropiată de cea folosită de grafcet: dreptunghiuri înnegrite subțiri.

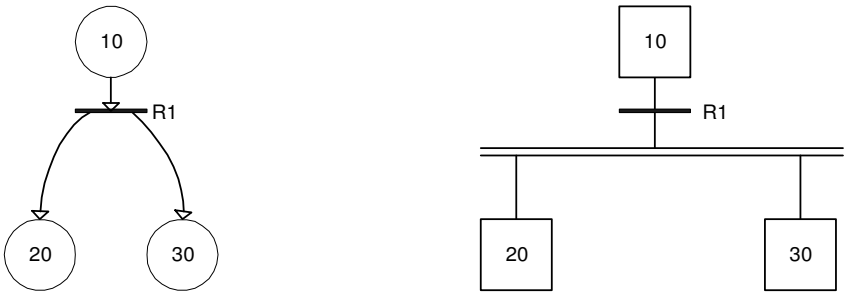


Fig. 4.13 Structura tip paralelism la rețeaua Petri a) şi grafcet b).

4.9.5. Reguli de evoluție în grafcet.

- **Regula 1. Condiții inițiale**. La momentul inițial numai etapele inițiale sunt active.
- **Regula 2. Condiții de validare.** Pentru ca o tranziție să fie validată este necesar ca toate etapele precedente să fie active.
- **Regula 3. Condiții de declanşare.** O tranziție este declanşată dacă este validată şi receptivitatea care o interpretează a apărut. Pentru a declanşa o tranziție trebuie, obligatoriu în această ordine, ca etapele precedente să fie dezactivate iar etapele posterioare să fie activate.
- **Regula 4. Declanşării simultane.** Toate tranzițiile declanşabile la un moment dat sunt declanşate simultan.
- **Regula 5. Conflictul de activare**. Dacă o etapă trebuie simultan activată şi desactivată de declanşările simultane ale unei tranziții precedente şi ale unei tranziții posterioare, atunci ea rămâne activă[117].

4.9.6. Structuri folosite la modelarea cu grafcet a sistemelor discrete logice.

Structurile tip folosite în grafcet sunt aceleaşi ca la rețeaua Petri 4.4.1: secvența (*Fig. 4.10* a), saltul (*Fig. 4.10* b), repetarea (*Fig. 4.10* c), alegerea (*Fig. 4.11* b) convergența (*Fig. 4.12* b), paralelismul (*Fig. 4.13* b) şi sincronizarea (*Fig. 4.14* b). După cum se vede din aceste figuri şi (*Fig. 4.7*) reprezentarea grafică este diferită.

O problemă apare la structura tip alegere (*Fig. 4.11*). La rețeaua Petri această structură realizează o alegere tip SAU EXCLUSIV iar la grafcet realizează o alegere tip SAU. Deosebirea apare atunci când evenimentele R1 şi R2 apar simultan. La grafcet, dacă etapa 10 este activă, se declanşează tranzițiile pe care aceste evenimente le interpretează si etapele 20 şi 30 devin active simultan, în *paralel*. Rețeaua Petri nu permite aceasta, ca o alegere să se transforme într-un paralelism! Diferența apare din modul diferit de definire a mărcii.

[117] Prin analogie cu pornirea unui motor electric se poate spune că există o prioritate la pornire.

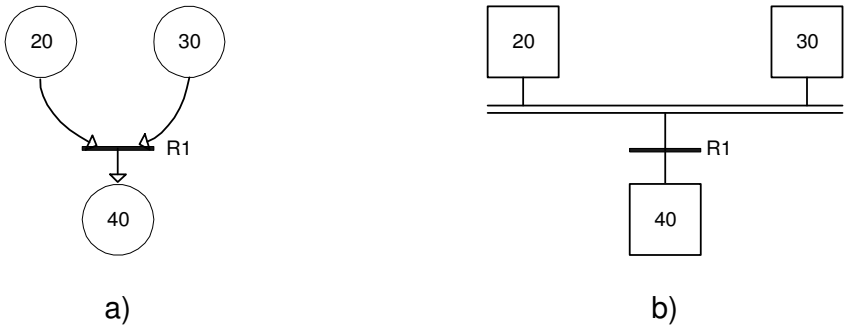


Fig. 4.14 Structura tip sincronizare la rețeaua Petri a) şi grafcet b).

La rețeaua Petri marca este o resursă şi este indivizibilă. Nu se poate deplasa în acelaşi timp din poziția 10 în 20 si 30. Din cauza aceasta apare conflictul efectiv, prezentat în paragraful 4.4.7, care trebuie rezolvat. Dacă nu este rezolvat sistemul discret logic modelat de rețeaua Petri se va comporta nedeterminist, la apariția simultană a evenimentelor R1 şi R2 se va activa etapa 20 sau (exclusiv) etapa 30 în funcție de factori care nu sunt cunoscuți.

La grafcet marca este un indicator, semnalizator al stării de activitate sau inactivitate a etapei. Deoarece mai multe etape pot fi active simultan şi marca se poate deplasa simultan din etapa 10 în etapele 20 şi 30.

4.9.7. Compararea grafcetului cu rețeaua Petri.

● *Avantajele* grafcetului față de rețeaua Petri sunt următoarele:

Grafcetul permite interpretarea tranzițiilor şi în funcție de unele evenimente interne (endogene) cum ar fi, de exemplu, apariția stării *active* sau *inactive*[118] a pozițiilor. Drept urmare desenul, care are o reprezentare grafică standardizată internațional, este mult mai simplu şi mai inteligibil.

Existența macro-etapelor în grafcet conduce de asemenea la o simplificare substanțială a caietelor de sarcini şi programelor pentru automatele programabile logice.

Grafcetul permite declanşarea simultană a tranzițiilor fără să existe noțiunea de conflict. Aceasta uşurează întocmirea şi folosirea caietelor de sarcini de către nespecialişti. O poziție activată şi desactivată simultan rămâne în grafcet activă[119]. In felul acesta se evită ignorarea acțiunilor corespunzătoare poziției şi modificarea algoritmului care a stat la baza întocmirii grafcetului. Ca un preț pentru facilitățile oferite, grafcetul nu posedă un model matematic atât de elaborat ca rețeaua Petri.

Desenul standardizat al grafcetului este mai adecvat pentru automatele programabile logice.

---

[118] Rețeaua Petri nu poate testa dacă o poziție este inactivă.

[119] Prioritate la pornire.

● *Dezavantajele* grafcetului față de rețeaua Petri sunt următoarele:

Grafcetul permite şi paralelismul prin interpretarea tranzițiilor situate într-o structură tip alegere. În aceste situații pot apare frecvent greşeli la ieşirea din paralelism.

La modelarea sistemelor discrete logice cu ajutorul grafcetului pot apare erori la reactivarea etapelor. La modelul rețea Petri aceste erori sunt repede evidențiate.

Structura modelului grafcet fără interpretare oferă puține informații şi nu permite verificările care se pot face pe un model rețea Petri.

In concluzie se poate aprecia că pentru elaborarea caietelor de sarcini sau programarea automatelor programabile logice grafcetul este preferabil. Pentru analiza, sinteza şi validarea modelului este mai bună rețeaua Petri. La prima vedere ar părea avantajoasă elaborarea caietului de sarcini sub formă de grafcet şi apoi să se treacă la rețeaua Petri pentru analiză, validare şi sinteză. Din păcate trecerea grafcet – rețea Petri şi invers nu se poate efectua direct decât în cazul modelelor mai simple de tip rețea Petri interpretată, sigură, viabilă şi fără conflicte structurale. In acest caz se renunță însă la toate facilitățile pe care le oferă grafcetul. De fapt alegerea ar trebui să se facă în funcție de complexitatea sistemului modelat. Pentru sisteme relativ simple nu are importanță modelul folosit. La modelarea sistemelor complexe grafcetul este bun drept caiet de sarcini, dar pentru analiza, sinteza şi validarea modelului este prudent să se folosească rețeaua Petri. In felul acesta greşeli greu detectabile pot fi evitate.

Grafcetul are o serie de caracteristici care îl deosebesc de rețeaua Petri şi îl fac deosebit de util în multe circumstanțe, dintre care se remarcă în special elaborarea de caiete de sarcini pentru sisteme discrete logice complexe şi programarea automatelor programabile logice. Astfel grafcetul permite interpretarea tranzițiilor şi în funcție de unele evenimente interne (endogene) cum ar fi, de exemplu, apariția stării active sau inactive[120] a pozițiilor. Drept urmare desenul, care are o reprezentare grafică standardizată internațional, este mai simplu şi mai inteligibil. Grafcetul permite declanşarea simultană a tranzițiilor fără să existe noțiunea de conflict. Aceasta uşurează întocmirea şi folosirea caietelor de sarcini de către nespecialişti. O poziție activată şi desactivată simultan rămâne în grafcet activă[121], spre deosebire de rețeaua Petri. In felul acesta se evită ignorarea acțiunilor corespunzătoare poziției şi modificarea algoritmului care a stat la baza întocmirii grafcetului. Ca un preț pentru facilitățile oferite, grafcetul nu posedă un model matematic atât de elaborat ca rețeaua Petri.

[120] Rețeaua Petri nu poate testa dacă o poziție nu este activă.

[121] Prioritate la pornire.

4.10. Sinteza sistemelor discrete logice.

Rezolvarea problemei sintezei sistemelor discrete logice constă în determinarea *relațiilor logice*. Acestea sunt de două tipuri. Primul tip de relații logice determină variabilele de stare prin funcții logice de timp şi de evenimentele de intrare. Al doilea tip de relații logice determină variabilele de ieşire prin funcții logice de variabilele de stare şi de evenimentele de intrare. Dacă la sistemele logice combinaționale sinteza se făcea pe baza tabelului de adevăr, la sistemele discrete logice sinteza se face pe baza modelului tip *rețea Petri conformă* sau tip *grafcet*. Deoarece cele două tipuri de modele sunt, după cum s-a arătat, identice în cele mai multe din cazurile întâlnite frecvent în practică, în continuare modelul sistemului discret logic va fi denumit simplu *rețea Petri*.

Există multe metode de sinteză a sistemelor discrete logice. În această lucrare se va pun accent în special pe proiectarea sistemelor discrete logice tip automat finit implementate cu contacte şi relee sau a sistemelor discrete logice tip grafcet implementate cu automate programabile logice.

Prima etapă a sintezei constă în întocmirea caietului de sarcini. În funcție de complexitatea sistemului caietul de sarcini poate fi o rețea Petri sau un grafcet. În ambele cazuri este recomandabil să se tracă la rețeaua Petri echivalentă care poate fi analizată mai uşor şi permite descoperirea multor erori care nu pot fi evidențiate pe grafcet.

Urmează, în a doua etapă a sintezei, efectuarea *analizei structurale şi comportamentale* a caietului de sarcini. Dacă acesta este sub formă de grafcet şi urmează să fie implementat pe automate programabile logice, se preferă uneori analiza prin simulare după implementare. Această atitudine se datorează faptului ca grafcetul se poate implementa automat[122]. Totuşi analiza pe rețeaua Petri, inclusiv prin simulare, este mult mai sigură în privința evitării unor tipuri de erori.

Rețeaua Petri se *codifică* în etapa următoare a sintezei. Dacă rețeaua este de tip automat finit şi se implementează cu contacte şi relee se foloseşte codul Gray. Dacă rețeaua este de tip grafcet, se implementează pe automat programabil logic şi nu se foloseşte facilitatea de programarea automată se face o codificare distribuită, tip 1 din n.

Relațiile logice se determină în funcție de tipul implementării. Dacă se folosesc contacte şi relee se foloseşte diagrama Karnaugh cu variabile incorporate. Dacă implementarea se face pe un automat programabil logic[123] se folosesc un grup de ecuații care modelează grafcetul.

---

[122] Implementarea automată este evitată de unii profesioniști. Întocmirea unui program pentru un automat programabil logic nu este aşa de dificilă sau laborioasă şi poate conduce la rezultate mult mai bune decât programarea automată. In plus, dacă sistemul nu funcționează bine se ştie foarte bine cine este vinovatul!

[123] In cazul în care nu se doreşte sau nu se poate face programarea automată direct pe baza grafcetului.

4.11. Implementarea sistemelor discrete logice.

După sinteză se face implementarea sistemelor discrete logice. Pentru aceasta se folosesc relațiile logice stabilite, relațiile fizice ale componentelor cu care se face implementarea şi relațiile de echivalență între variabilele. Deoarece sinteza depinde, în oarecare măsură, de tipul de implementare ales, cele două probleme vor fi rezolvate împreună. În continuare se vor considera implementarea cu contacte şi relee, cu circuite integrate logice şi cu automat programabil. Implementările cu contacte şi relee sau cu automat programabil sunt cele mai frecvente în domeniul conducerii instalațiilor pe baza unor modele discrete logice. Implementarea cu circuite integrate logice este prezentată pentru a înțelege mai bine funcționarea şi programarea automatului programabil logic şi pentru a aprecia avantajele şi dezavantajele celorlalte două tipuri de implementări .

4.12. Automate elementare cu contacte şi releu.

Cele mai simple sisteme discrete logice au numai două stări şi din această cauză se numesc adeseori *elementare* sau *bistabile*. Rețelele viabile[124] care modelează aceste sisteme sunt în număr de patru şi sunt prezentate în primul rând din Fig. 4.15. Alte combinații de arce între cele două poziții (stări) mai sunt posibile, dar rețelele obținute sau pot fi reduse la cele patru tipuri prezentate sau nu mai sunt viabile şi prin urmare nu pot modela sisteme discrete logice reale. Am codificat denumirea celor două poziții cu ajutorul variabilei de stare *x*. Poziția inițial activă ***A*** are variabila de stare *x=0* iar cea de-a doua poziție ***B*** este codificată cu variabila de stare *x=1*, după cum se vede în Fig. 4.15.

Aplicația cea mai răspândită[125] a unora dintre aceste bistabile, în varianta implementării cu contacte şi relee, o constitue automatele pentru pornirea directă a motoarelor electrice. Din această cauză am dat variabilelor care interpretează pozițiile şi tranzițiile denumiri legate de această aplicație.

Pozițiile rețelei sunt interpretate de variabila de ieşire *y*. Valoarea *y=0* a variabilei de ieşire corespunde comenzii de motor oprit, iar valoarea *y=1* a variabile de ieşire corespunde comenzii de motor pornit.

Evenimentele care interpretează tranzițiile α, β, γ, δ sunt acționarea butoanelor de pornire *p* şi respectiv oprire *o*.

---

[124] Fără blocări.

[125] Alte aplicații sunt de asemenea frecvente, de exemplu regulatorul bipozițional sau blocul de memorare din cadrul unor sisteme discrete logice mai complexe.

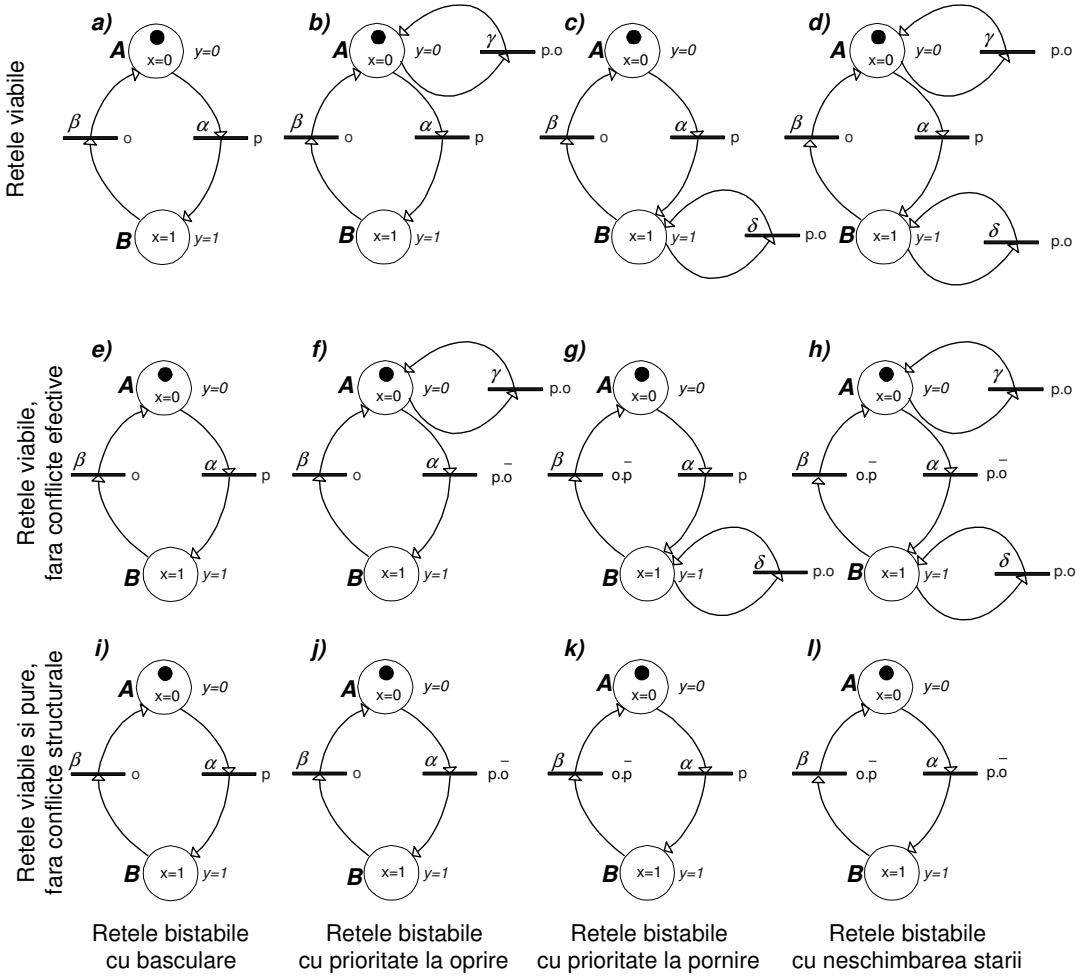


Fig. 4.15 Analiza structurală a celor patru variante posibile de rețele bistabile viabile. Transformarea lor în rețele viabile şi pure, fără conflicte structurale.

4.12.1. Analiza structurală.

Analiza structurala a rețelelor viabile din Fig. 4.15 b,c,d) indică existența unor conflicte structurale. De exemplu, la rețeaua din Fig. 4.15 b) din poziția inițială ***A*** pleacă două arce. Unul către tranziția α şi altul către tranziția *γ.* Aceste conflicte devin şi efective atunci când se acționează simultan butoanele de pornire *p* şi oprire *o*. Eliminarea conflictelor efective prin interpretare este prezentată în Fig. 4.15 f,g,h). De exemplu, conflictul din Fig. 4.15 b) a fost eliminat interpretând altfel tranziția *α.* În loc ca aceasta să se declanşeze la apăsarea butonului *p* ea se va declanşa, ca în Fig. 4.15 f), atunci când se apasă butonul *p* şi nu se apasă butonul *o*. În felul acesta, la apăsarea simultană a butoanelor *p* şi *o* se declanşează o singură tranziție γ, cea interpretată de evenimentul *p.o*.

Conflictele efective au dispărut, dar rețelele din Fig. 4.15 f,g,h) sunt în continuare degenerate pentru că au arce care pleacă şi se întorc din aceiaşi poziție. Se observă însă că prin

interpretarea făcută tranzițiilor, pentru eliminarea conflictelor efective, aceste arce nu mai sunt necesare şi pot fi eliminate. Într-adevăr, aceste arce indică faptul că la apăsarea simultană pe butoanele de pornire şi oprire *p* şi *o*, adică atunci când *p.o=1*, pozițiile din care pleacă şi se întorc arcele rămân active. Dacă eliminăm aceste arce, la apariția evenimentului *p.o=1* se observă că, datorită interpretării făcute, pozițiile respective rămân active.

4.12.2. Analiza comportamentală.

Toate rețelele viabile şi pure, fără conflicte structurale din ultimul rând al Fig. 4.15 au următoarea comportare.

În poziția ***A,*** activă inițial, variabila de stare este *x=o,* variabila de ieşire este *y=0,* şi motorul este oprit. În această situație este validată numai tranziția *α.*

Dacă evenimentul care constă din apăsarea butonului de pornire apare, adică *p=1*, atunci tranziția α care este interpretată de acest eveniment se declanşează, poziția precedentă ***A*** se dezactivează, iar poziția posterioară ***B*** se activează. În consecință *y=1* iar motorul porneşte. În această nouă situație în care s-a ajuns este validată numai tranziția β.

Atunci când se apasă butonul de oprire variabila logică *o=1* şi tranziția β pe care o interpretează se declanşează deoarece este validată. Iarăşi poziția precedentă ***B*** se dezactivează iar poziția posterioară ***A*** se activează şi motorul se opreşte. S-a ajuns în situația de la care s-a pornit, deci rețelele analizate realizează performanța de *reversibilitate.*

În urma analizei comportamentale se observă că rețelele nu se blochează, deci au performanța de *viabilitate* şi nu au niciodată mai mult de o marcă în fiecare poziție, deci au performanța de *siguranță*.

La apăsarea simultană a butoanelor de pornire şi oprire, adică atunci când *p.o=1* comportarea rețelelor din ultimul rând al Fig. 4.15 nu mai este aceiaşi.

Pentru rețeaua din Fig. 4.15 i) la această manevră se produce o basculare. Dacă era activă poziția **A** cu *x=0* după manevră devine activă poziția ***B*** cu *x=1* şi invers. Din această cauză rețeaua este denumită bistabilă cu basculare.

Pentru rețeaua din Fig. 4.15 j) la apăsarea simultană a butoanelor de pornire şi oprire devine activă poziția ***A*** cu *x=0* şi motorul se opreşte, indiferent în ce situația se face această manevră, adică indiferent care poziție este activă. Se constată deci o prioritate a comenzii de oprire şi din această cauză rețeaua se numeşte bistabilă, cu prioritate la oprire.

Rețeaua din Fig. 4.15 l) are o comportare asemănătoare cu rețeaua precedentă, numai că la apăsarea simultană a butoanelor de pornire şi oprire *p.o=1* motorul porneşte, indiferent de starea în care se găseşte, adică indiferent de poziția ***A*** sau ***B*** care este activă. În consecință rețeaua este denumită bistabilă cu prioritate la pornire.

În sfârşit, pentru sistemul logic discret modelat de rețeaua din Fig. 4.15 m) apăsarea simultană pe butoanele de pornire şi oprire *p.o=1* nu provoacă nici o modificare. Dacă motorul era oprit, adică poziția ***A*** este activă, rămâne oprit şi dacă era pornit, adică poziția ***B*** este activă, rămâne pornit. Rețeaua se numeşte bistabilă cu neschimbarea stării.

4.12.3. Automatul elementar cu basculare.

Schema bloc a automatului elementar cu basculare este prezentată în Fig. 6.1 b), iar modelul sub forma unei rețele Petri conforme în Fig. 4.15 i). Din schema bloc se observă că variabilele de intrare sunt *p* şi *o*, iar variabile de ieşire este *y*. Din modul de funcționare stabilit la analiza comportamentală efectuată în paragraful precedent rezultă că ieşirea depinde de intrări prin intermediul variabilei de stare *x*. Pentru codificarea pozițiilor din Fig. 4.15 i)[126] rezultă următoarea relație dintre ieşire şi stare[127]:

$$y = x \tag{4.35}$$

Deci ieşirea este identică cu starea. Din rețeaua Petri prezentată în Fig. 4.15 i) rezultă că starea automatului depinde de cele două intrări. *p* şi *o*, dar şi de starea precedentă a automatului. De exemplu, pentru următoarele intrări *p=1* şi *o=1* starea automatului basculează. Dacă înainte de aplicarea intrărilor era *x=0*, după aplicarea intrărilor devine *x=1*. Din (4.35) rezultă că şi ieşirea basculează.

Pentru întocmirea tabelului de adevăr al automatului cu basculare trebuie să se iau în considerare toate valorile intrărilor p^t şi o^t şi a stărilor x^t la momentul *actual de timp t* şi se determină pe baza rețelei din Fig. 4.15 i) stările $x^{t+\Delta}$ la momentul *viitor de timp* $t+\Delta$. În coloanele din stânga ale Fig. 4.16 sunt prezentate toate valorile posibile ale variabilelor de intrare *p* şi *o* şi ale variabilei de stare *x* la momentul de timp prezent *t.* Pentru completarea tabelului de adevăr, pe baza rețelei Fig. 4.15 i), se porneşte întotdeauna de la starea actuală[128], dată de valoarea variabilei de stare x^t din coloana a treia a tabelului. Să completăm pe rând liniile tabelului de adevăr din Fig. 4.16.

În prima linie starea actuală este $x^t=0$, adică ne situăm în poziția ***A,*** activă inițial a rețelei din Fig. 4.15 i) şi deci *y=0,* iar motorul este oprit. Deoarece variabilele de intrare din prima linie a Fig. 4.16 sunt zero, adică $p^t=0$ şi $o^t=0$, înseamnă că nu se apasă nici un buton. Prin urmare nu se declanşează nici o tranziție a rețelei şi, la momentul de timp imediat următor $t+\Delta$ rămâne activă tot poziția inițială ***A*** cu $x^{t+\Delta}=0$ şi motorul oprit. Se trece valoarea în ultima coloană a tabelului.

---

[126] Adică poziției ***A*** îi corespunde valoarea variabilei de stare *x=0*, iar poziției ***B*** îi corespunde valoarea variabilei de stare *x=1*.

[127] Alte relații sunt posibile. De exemplu în cazul în care poziția inițială este codificată cu *x=1,* iar cealaltă poziție este codificată cu *x=0,* relația dintre ieşire şi stare este $y = \overline{x}$

[128] Starea prezentă, starea la momentul de timp *t*.

p^t	o^t	x^t	$x^{t+\Delta}$
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**0**
1	0	0	**1**
1	0	1	**1**
1	1	0	**1**
1	1	1	**0**

Fig. 4.16 Tabelul de adevăr pentru automatul elementar cu basculare.

Linia a doua a Fig. 4.16indică că starea inițial activă a rețelei este poziția ***B*** cu $x^t=1$ şi motorul este pornit. În această linie variabilele de intrare sunt zero, adică nu apare nici un eveniment şi deci nu se declanşează nici o tranziție a rețelei şi prin urmare rămâne activă tot poziția ***B*** corespunzătoare motorului pornit. Deci la momentul de timp imediat următor $x^{t+\Delta}=1$ şi se completează în consecință ultima coloană a Fig. 4.16.

În linia a treia şi coloana a treia avem $x^t=0$, deci este activă poziția ***A*** şi motorul este oprit. Se apasă numai butonul de oprire, adică $p=0$ şi $o=1$. Apariția evenimentului $o=1$ nu declanşează însă tranziția β pe care o interpretează deoarece aceasta nu este validată. Pentru a fi validată este necesar să fie activă poziția ***B*** cu $x^t=1$. Prin urmare rămâne activă tot poziția ***A*** şi la momentul de timp imediat următor $x^{t+\Delta}=0$ iar motorul rămâne oprit.

Valorile de intrare ale liniei a patra din Fig. 4.16 indică că este activă poziția ***B*** ($x^t=1$), motorul este pornit şi se apasă butonul de oprire $o^t=1$. Din modelul automatului cu basculare din Fig. 4.15 i) rezultă că se declanşează tranziția validată β pe care o interpretează evenimentul *o*, deci poziția precedentă ***B*** se dezactivează iar poziția posterioară ***A*** se activează. Se trece în coloana a patra a tabelului valoarea $x^{t+\Delta}=0$. *M*otorul se opreşte.

O situație clasică apare în linia cinci a tabelului. Motorul este oprit şi se apasă pe butonul de pornire. Evident, motorul ar trebui să pornească. Dar să vedem dacă aceasta se întâmplă şi atunci când analizăm comportamental rețeaua conformă din Fig. 4.15 i). Pentru această linie a tabelului este activă poziția ***A***. Tranziția β este validată. Atunci când se apasă butonul de pornire apare evenimentul $p=1$ şi tranziția β se declanşează. Ca urmare a declanşării poziția ***A*** precedentă tranziției de dezactivează iar poziția ***B*** se activează. Deoarece poziția ***B*** este interpretată de variabila *y* cu valoarea 1 motorul porneşte.

Dacă motorul este pornit şi se apasă butonul de pornire, ca în linia şase, motorul trebuie să rămână pornit. Modelul tip rețea Petri conformă din Fig. 4.15 i) arată tocmai această comportare. În ultima coloană a tabelului de adevăr se trece deci valoarea $x^{t+\Delta}=1$.

Apăsarea simultană a butoanelor de pornire şi oprire, adică cazul în care $p=1$ şi $o=1$ este analizată în liniile şapte şi opt ale tabelului de adevăr. Rețeaua Petri arată că în acest caz

se produce o basculare, care de altfel a şi dat numele automatului elementar. Dacă era activă poziția ***A*** ($x^t=0$), cu motorul oprit, la apariția evenimentelor (apăsarea butoanelor) se declanşează tranziția α şi devine activă poziția ***B,*** caracterizată de valoarea variabilei de stare $x^{t+\Delta}=1$, şi motorul porneşte.

Situația se inversează în linia opt a tabelului. Este activă poziția ***B*** şi prin apăsarea simultană a butoanelor devine activă poziția ***A.*** Se trece $x^{t+\Delta}=0$ în tabel.

$x^{t+\Delta}$

x^t \ $p^t o^t$	00	01	11	10
0	0	0	1	1
1	1	0	0	1

Fig. 4.17 Diagrama Karnaugh pentru automatul elementar cu basculare

Se observă că tabelul de adevăr din Fig. 4.17 şi reţeaua Petri din Fig. 4.15 i) sunt echivalente.

Sinteza relațiilor logice pentru automatul elementar cu basculare se face cu ajutorul diagramei Karnaugh din Fig. 4.17obținute din tabelul de adevăr. Relația logică pentru $x^{t+\Delta}$ se obține la fel ca şi în cazul sistemelor logice combinaționale[129]. Cu grupările din Fig. 4.17 se obține relația (4.36).

Implementarea cu contacte şi releu a automatului elementar cu basculare se face considerând relația fizică[130] a releului (4.37) şi relația de echivalență (4.38) între variabila de stare *x* şi variabila de ieşire a releului, contactul său *k*.

$$x^{t+\Delta} = p^t.\overline{x^t} + \overline{o^t}.x^t \qquad (4.36)$$

$$k^{t+\Delta} = K^t \qquad (4.37)$$

$$x^t = k^t \qquad (4.38)$$

Între relațiile (4.36), (4.37) şi (4.38) se elimină variabilele *x* şi Δ. Se obține relația (4.39). În această relație toate variabilele au valoarea de la momentul de timp actual *t* şi din această cauză se poate renunța la specificarea timpului ca în (4.40).

[129] Se observă că sinteza sistemelor discrete logice se reduce la sinteza sistemelor logice combinaționale.

[130] Această relație a fost prezentată în capitolul precedent.

$$K^t = p^t.\overline{k^t} + \overline{o^t}.k^t \tag{4.39}$$

$$K = p.\bar{k} + \bar{o}.k \tag{4.40}$$

Implementarea cu contacte şi releu a relației logice (4.40) este prezentată în Fig. 4.20. Contactul de ieşire *k* al automatului acționează un semnalizator. Această schemă este mai puțin folosită în practică. Se preferă implementarea automatului cu basculare în varianta cu circuite logice integrate.

$\mathbf{p^t}$	$\mathbf{o^t}$	$\mathbf{x^t}$	$\mathbf{x^{t+\Delta}}$
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**0**
1	0	0	**1**
1	0	1	**1**
1	1	0	**0**
1	1	1	**0**

Fig. 4.18 Tabelul de adevăr pentru automatul elementar cu prioritate la oprire.

Legăturile inverse din Fig. 4.20 leagă releul K cu contactele sale k situate la intrarea sa. Când se apasă butonul de pornire *p* releul K este acționat prin circuitul 10 şi după intervalul de timp Δ îşi acționează contactul său k din circuitul 20 care se închide. Atunci când butonul de pornire *p* nu mai este acționat circuitul 10 se întrerupe dar releul rămâne acționat prin circuitul 20 deoarece contactul său *k* este închis. Din această cauză contactul *k* mai este denumit *contact de automenținere*. Deoarece contactul *k* din circuitul 20 are o acțiune care se adună al acțiunea butonului *p,* legătura inversă corespunzătoare se numeşte *pozitivă*.

Circuitul 10 de acționare a releului este întrerupt nu numai de încetarea acționării butonului *p* ci şi de acționarea contactului *k* normal închis al releului. Deoarece acțiunea acestui contact se opune se opune intrării în releu care tinde să în acționeze, legătura inversă corespunzătoare se numeşte negativă.

În analiza efectuată mai înainte s-a presupus că ambele contacte *k* ale releului din circuitele 10 şi 20 sunt acționate cu exact aceiaşi întârziere Δ. Dacă întârzierile sunt diferite, ceea ce este cazul în practică, apare fenomenul de *hazard combinațional*. Rezolvarea diferitor aspecte legate de parametrii fizici ai componentelor sistemului este tratată în capitolul consacrat proiectării sistemelor discrete logice. Sinteza şi implementarea sistemelor discrete logice se referă numai aspectele funcționale ale sistemului şi presupun caracteristici ideale pentru componentele sale.

4.12.4. Automatul elementar cu prioritate la oprire.

Sinteza şi *implementarea* automatului elementar cu prioritate la oprire se face la fel ca şi în cazul automatului elementar cu basculare din paragraful precedent .

$x^{t+\Delta}$

x^t \ $p^t o^t$	00	01	11	10
0	0	0	0	1
1	1	0	0	1

Fig. 4.19 Diagrama Karnaugh pentru automatul elementar cu prioritate la oprire.

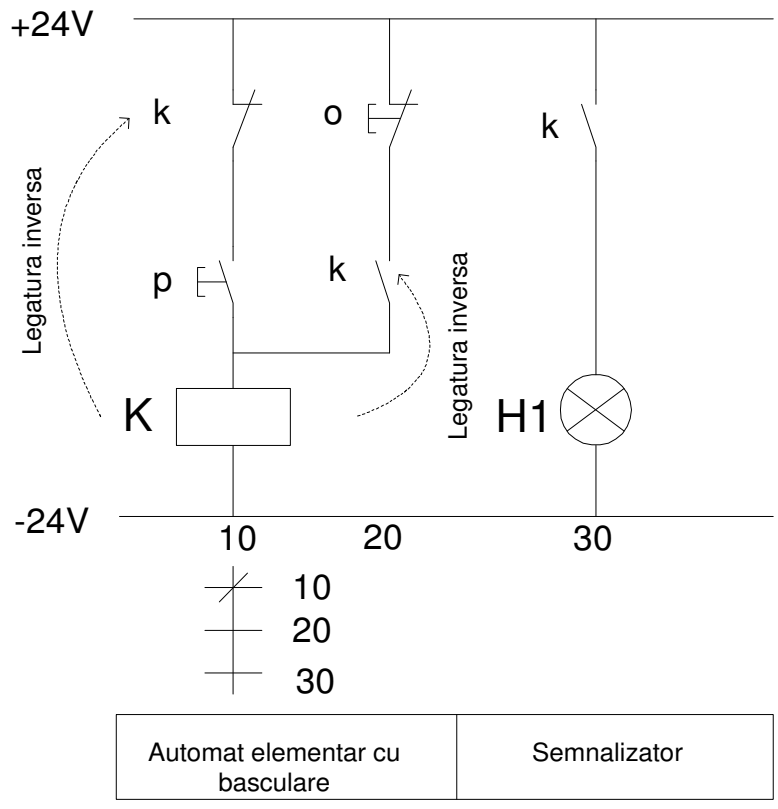


Fig. 4.20 Schema desfăşurată electrică a automatului elementar cu basculare.

Modelul automatului elementar cu prioritate la oprire este prezentat în Fig. 4.15 j). Analiza comportamentală a modelului permite completarea tabelului de adevăr din Fig. 4.18.

Modificarea care apare în acest caz se referă la liniile şapte şi opt ale tabelului de adevăr. Indiferent care este starea prezentă a reţelei, adică şi în cazul în care este activă poziţia ***A*** cu x^t=0, şi în cazul în care este activă poziţia ***B*** cu $x^t=1$, la apăsarea simultană a butoanelor de pornire ($p^t=1$) şi oprire ($o^t=1$), devine activă poziţia ***A*** cu $x^{t+\Delta}=0$. Această poziţie este interpretată de valoarea variabilei logice de ieşire $y=0$ şi în consecinţă motorul se opreşte. Denumirea automatului provine din faptul că la apăsarea simultană a butoanelor de pornire şi de oprire are *prioritate* butonul de oprire. Această manieră prudentă de tratare a evenimentelor concurente neprevăzute[131] este adoptată frecvent în practica inginerească.

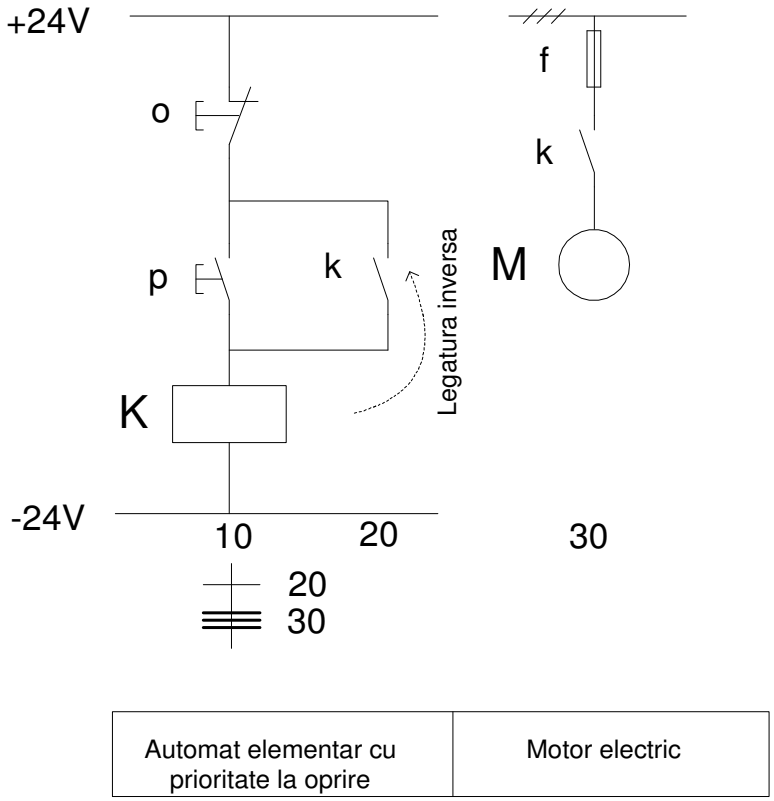


Fig. 4.21 Schema desfăşurată electrică pentru automatul elementar cu prioritate la oprire.

Diagrama Karnaugh corespunzătoare tabelului de adevăr din Fig. 4.18 este prezentată în Fig. 4.19. Pentru cele două grupări din diagrama Karnaugh se obţine relaţia logică (4.41).

$$x^{t+\Delta} = \overline{o^t}.p^t + \overline{o^t}.x^t = \overline{o^t}.(p^t + x^t) \qquad (4.41)$$

[131] Apăsarea simultană a butoanelor de pornire şi oprire este o manevră care nu este prevăzută de obicei mla funcţionarea normală a instalaţiilor. Din această cauză este mai prudent ca la această manevră instalaţia să se oprească.

Pentru implementarea cu contacte şi releu se procedează la fel ca în cazul automatului cu basculare prezentat în paragraful precedent. Se scriu relaţia fizică a releului şi relaţia de echivalare a variabilelor li se elimină variabilele *x, Δ* şi *t*. Se obţine relaţia (4.42) care se implementează ca în Fig. 4.21. Contactul de ieşire *k* al automatului acţionează în acest caz[132] un motor electric. Aceasta este o schemă desfăşurată electrică *simplificată*[133] care ilustrează o situaţie frecvent întâlnită în practică. Acest automat este folosit adeseori şi ca o componentă a unor sisteme mai complexe de conducere.

Automatul cu prioritate la oprire, la fel ca şi automatul cu basculare, prezintă o *legătură inversă*[134] *pozitivă* în circuitul 20. Contactul de automenţinere *k* din acest circuit ajută acţiunea butonului *p,* care constă în acţionarea releului K, şi o menţine şi după ce acţiunea acestuia a încetat. Ne reamintim definiţia sistemului automat: un sistem cu memorie sau legătură inversă. În cazul automatelor elementare legătura inversă realizează tocmai acţiunea de memorare[135]. Având în vedere această proprietate cu totul remarcabilă a legăturii inverse se poate afirma că pentru a construi un sistem automat este suficient să introducem legături inverse.

$$K = \bar{o}.(p + k) \qquad (4.42)$$

$\mathbf{p^t}$	$\mathbf{o^t}$	$\mathbf{x^t}$	$\mathbf{x^{t+\Delta}}$
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**0**
1	0	0	**1**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

Fig. 4.22 Tabelul de adevăr pentru automatul cu prioritate l pornire.

[132] Automatul cu prioritate la oprire poate acţiona şi alte echipamente, cum ar fi , de exemplu, o instalaţie electrică de încălzire.

[133] Unele elemente importante ale schemei clasice de pornire directa a unui motor, cum ar fi releul termic de protecţie a motorului, au fost omise, accentul punându-se pe sinteza şi implementarea automatului cu prioritate la oprire. Se observă alimentarea separată a părţii de comandă automată şi a părţii de forţă a instalaţiei.

[134] Feedback (eng.)

[135] Memorarea se poate realiza şi prin alte metode, diferite de metoda legăturii inverse, de exemplu folosind un sistem de came. Şi în prezent programul de acţiuni secvenţiale la majoritatea maşinilor automate de spălat este memorat şi se realizează cu ajutorul unui sistem de came.

4.12.5. Automatul elementar cu prioritate la pornire.

Spre deosebire de automatul din paragraful precedent, care se opreşte la apăsarea simultană a butoanelor de pornire şi oprire, acest automat *porneşte*. Se spune că are prioritate butonul de pornire. Tabelul de adevăr întocmit, la fel ca şi la , pe baza modelului sub formă de rețea Petri conformă (Fig. 4.15 k) este prezentat în Fig. 4.22.

$x^{t+\Delta}$

x^t \ $p^t o^t$	00	01	11	10
0	0	0	1	1
1	1	0	1	1

Fig. 4.23 Diagrama Karnaugh pentru automatul elementar cu prioritate la pornire.

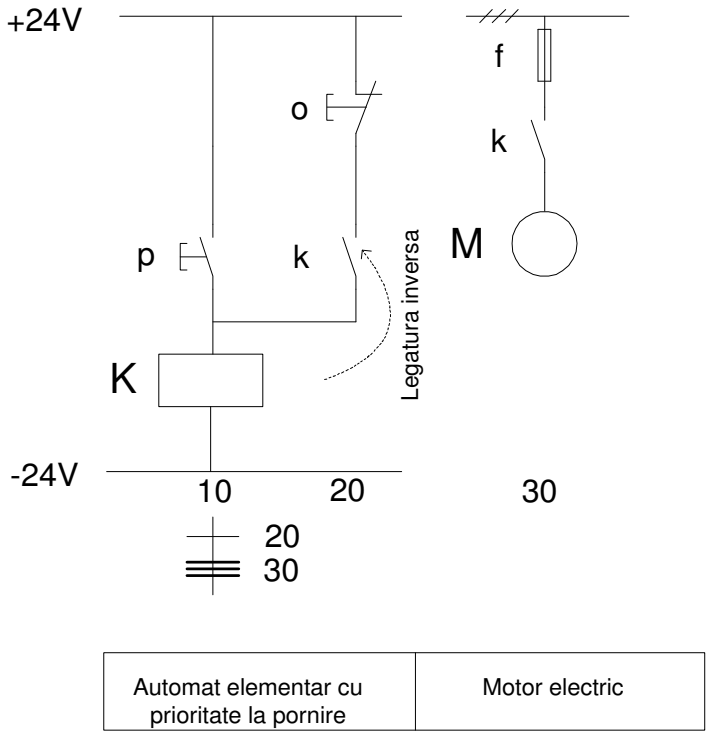


Fig. 4.24 Schema desfăşurată electrică pentru automatul elementar cu prioritate la pornire.

Alegerea priorității la pornire se face în liniile şapte şi opt ale tabelului. Dacă $x^t=0$, adică este activă poziția ***A*** şi motorul este oprit, este validată tranziția α rețelei din Fig. 4.15

k). La apariția evenimentelor simultane $p=1$ şi $o=1$ tranziția α se declanşează pentru că este validată şi interpretată de evenimentul p care a apărut. Poziția ***A***, precedentă tranziției α, se dezactivează şi poziția ***B***, posterioară tranziției α, se activează. Dacă examinăm rețeaua la momentul următor declanşării tranziției, adică la timpul $t+\Delta$, constatăm că singura poziție activă este ***B***. Variabila de starea a automatului ia acum valoarea $x^{t+\Delta}=1$. Deoarece în cazul rețelelor din Fig. 4.15 variabilele de stare sunt egale cu variabilele de ieşire avem şi $y^{t+\Delta}=x^{t+\Delta}=1$, adică motorul porneşte.

În ultima linie, a opta, a tabelului de adevăr variabila de stare s automatului cu are valoarea $x^t=1$, adică este activă poziția ***B*** a rețelei din Fig. 4.15 k) şi motorul este pornit. Deoarece ***B*** este activă, tranziția β este validată. Pentru rețeaua analizată tranziția β este interpretată de un eveniment care apare atunci când se apasă butonul de oprire dar nu se apasă butonul de pornire. În linia opt a tabelului de adevăr valorile variabilelor p şi o arată ca în această situație ambele butoane sunt apăsate. Deci nu este îndeplinită condiția de declanşare a tranziției β şi poziția ***B*** rămâne în continuare activă, cu motorul pornit.

$\mathbf{p^t}$	$\mathbf{o^t}$	$\mathbf{x^t}$	$\mathbf{x^{t+\Delta}}$
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**0**
1	0	0	**1**
1	0	1	**1**
1	1	0	**0**
1	1	1	**1**

Fig. 4.25 Tabelul de adevăr pentru automatul elementar cu neschimbarea stării.

Pentru sinteza automatului elementar cu prioritate la pornire datele din tabelul de adevăr Fig. 4.22 se trec în diagrama Karnaugh din Fig. 4.23. Se pot face două grupări şi rezultă următoarea relație logică.

$$x^{t+\Delta} = p^t + \overline{o^t}.x^t \qquad (4.43)$$

După considerarea relației fizice (4.37) a releului şi a relației de echivalență a variabilelor (4.38) rezultă relația (4.44) care poate fi implementată ca în Fig. 4.24.

$$K = p + \overline{o}.k \qquad (4.44)$$

Automatul cu prioritate la pornire, în varianta implementată cu contacte şi releu, este foarte puțin folosit în practică. În schimb acest automat este de bază la implementarea software pe baza caietului de sarcini grafcet.

Şi automatul cu prioritate la oprire îşi realizează acțiunea de memorare a comenzii de pornire a motorului dată prin butonul *p* cu ajutorul unei *legături inverse* realizate prin intermediul contactului de automenținere *k* din circuitul 20. Comparând schemele de la automatele cu prioritate la oprire şi cu prioritate la pornire se constată că în primul caz butonul de oprire întrerupe atât acțiunea butonului p cât şi acțiunea contactului de automenținere *k*, pe când în cel de al doilea caz oprirea întrerupe doar circuitul 20 de automenținere.

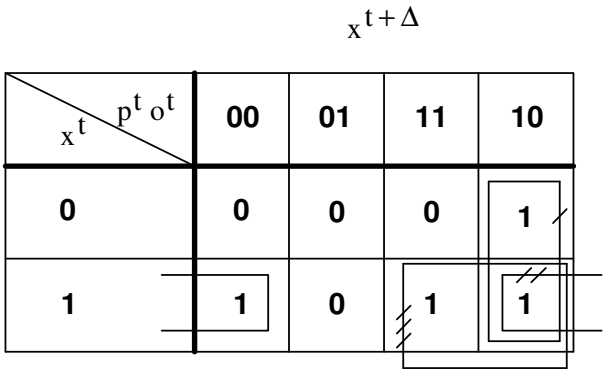


Fig. 4.26 Diagrama Karnaugh pentru automatul elementar cu neschimbarea stării.

4.12.6. Automatul elementar cu neschimbarea stării.

Acest automat nu şi-a găsit încă aplicarea în varianta implementată cu contacte şi releu. Se prezintă totuşi, ca un exercițiu, sinteza şi implementarea sa.

Paşii urmați sunt aceiaşi ca la celelalte automate elementare. Tabelul de adevăr din Fig. 4.25 este obținut prin analiza comportamentală a rețelei din Fig. 4.15 l). Se observă că pentru această rețea tranzițiile α şi β sunt astfel interpretate încât nu se declanşează decât atunci când se apasă un buton şi *nu* se apasă celălalt. În consecință, la apăsarea simultană a butoanelor de pornire şi oprire tranzițiile nu vor putea fi declanşate chiar dacă sunt validate. Deci la această manevră, apăsarea simultană a butoanelor, rețeaua şi automatul pe care îl modelează îşi păstrează starea. De aici provine şi denumirea automatului elementar. În tabelul de adevăr, pentru coloanele şapte şi opt avem $x^{t+\Delta}=x^t$.

Sinteza automatului se face trecând datele din tabelul de adevăr Fig. 4.25 în diagrama Karnaugh din Fig. 4.26 şi făcând grupări. Relația logică corespunzătoare celor trei grupări este dată de (4.45).

$$x^{t+\Delta} = p^t.\overline{o^t} + \overline{o^t}.x^t + p^t.x^t = \overline{o^t}.(p^t + x^t) + p^t.x^t \qquad (4.45)$$

Se adaugă relația fizică (4.37) şi relația de echivalență (4.38) fiind posibilă eliminarea variabilelor *x, Δ* şi *t.* Se obține relația logică finala (4.46)

$$K = \bar{o}.(p+k)+p.k \qquad (4.46)$$

Schema desfăşurată electrică corespunzătoare relației obținute prin sinteză este prezentată în Fig. 4.27.

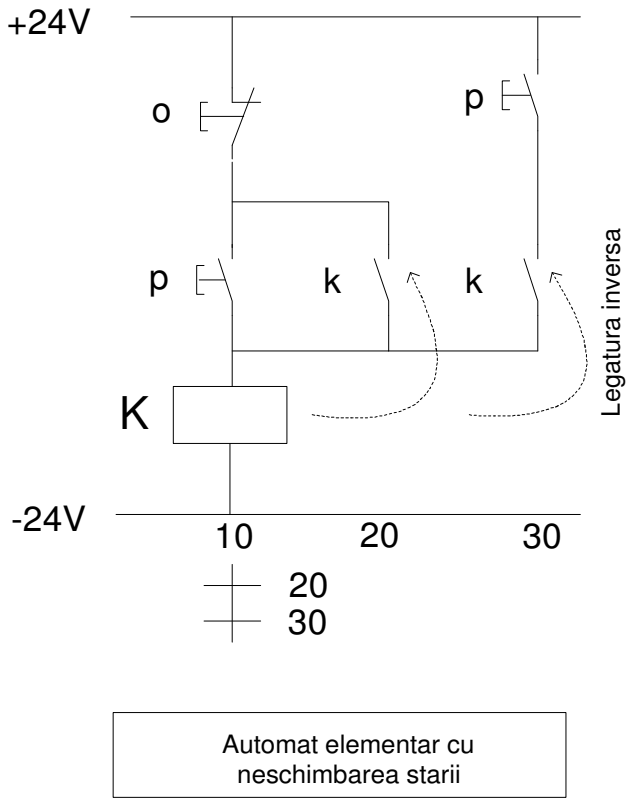


Fig. 4.27 Schema desfăşurată electrică pentru automatul elementar cu neschimbarea stării.

La automatul cu neschimbarea stării există de asemenea două *legături inverse* ca şi la automatul cu basculare. La apăsarea simultană a butoanelor p şi o automatul ar trebui să nu-şi schimbe starea. Acest lucru se şi întâmplă. Dar dacă acțiunea celor două butoane nu este perfect simultană pot apare erori.

4.12.7. Automatul elementar pentru reglarea bipoziţională.

Automatele elementare implementate cu contacte şi releu sunt aplicate frecvent la conducerea sistemelor. Un exemplu îl reprezintă automatul pentru reglarea bipoziţională a nivelului într-un rezervor prezentat în capitolul 2. În continuare să vedem cum poate fi sintetizat şi implementat.

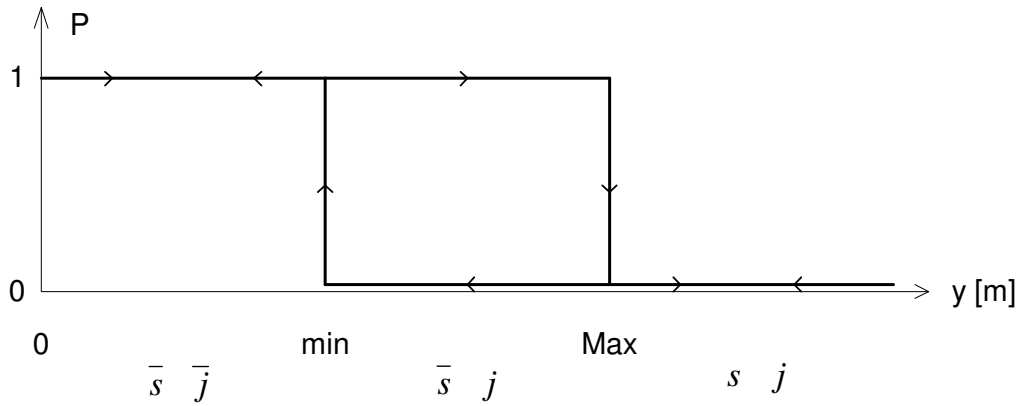


Fig. 4.28 Caracteristica de funcţionare a pompei instalaţiei din Fig. 2.7.

Schema tehnologică a sistemului automat este prezentată în Fig. 2.7 iar schema bloc în *Fig. 2.9*. Caietul de sarcini al automatului discret logic care conduce instalaţia este arătat în cuvinte în capitolul 2 sau, sub formă de algoritm, în Lista 2.1. Pe baza acestui caiet de sarcini se poate proiecta automatul printr-o metodă oarecare, de exemplu pe bază de exemple tip, rezultând schema desfăşurată din Fig. 2.8, în varianta implementată cu contacte şi relee, sau schema desfăşurată electrică din *Fig. 2.12*, în varianta implementată cu automat programabil logic. În acest paragraf se prezintă sinteza şi implementarea cu contacte şi releu a automatului pe baza modelului sub forma unei reţele Petri conforme.

Reglarea automată a nivelului în rezervor se face, în exemplul considerat, pe baza unui algoritm bipoziţional care porneşte sau opreşte pompa în funcţie de nivelul măsurat în rezervor. Traductoarele sistemului au la ieşire un contact *s* care este acţionat *(s=1)* atunci când nivelul *y* depăşeşte valoarea **Max** şi un contact *j* care este acţionat *(j=1)* atunci când nivelul *y* depăşeşte valoarea **min.** Caracteristica de funcţionare a pompei, conform acestui algoritm, este prezentată în Fig. 4.28.

În graficul din Fig. 4.28 se disting trei zone. Pentru zona în care nivelul $y \geq Max$ contactele *s* şi *j* sunt acţionate[136] *(s=1, j=1)*, iar pompa este oprită *(P=0).* În zona în care nivelul $y < min$ contactele *s* şi *j* **nu**[137] sunt acţionate *(s=0, j=0),* iar pompa este pornită *(P=1).* Zona intermediară caracterizată de nivelul $min \leq y < Max$ este sesizată de traductoare astfel:

[136] Evenimentele *a* şi *b* au apărut, în exprimarea specifică reţelelor Petri conforme.

[137] În Fig. 4.28 evenimentele a şi b sunt negate, indicând faptul că nu au apărut.

contactul a nu este acționat *(s=0)* şi contactul b este acționat *(j=1).* În *această zonă* pompa este pornită sau oprită în funcție de starea sa precedentă. Dacă era pornită rămâne pornită şi dacă era oprită rămâne oprită. Caietul de sarcini al automatului care porneşte şi opreşte pompa conform acestui algoritm este prezentat sub forma unei rețele Petri în Fig. 4.29 a).

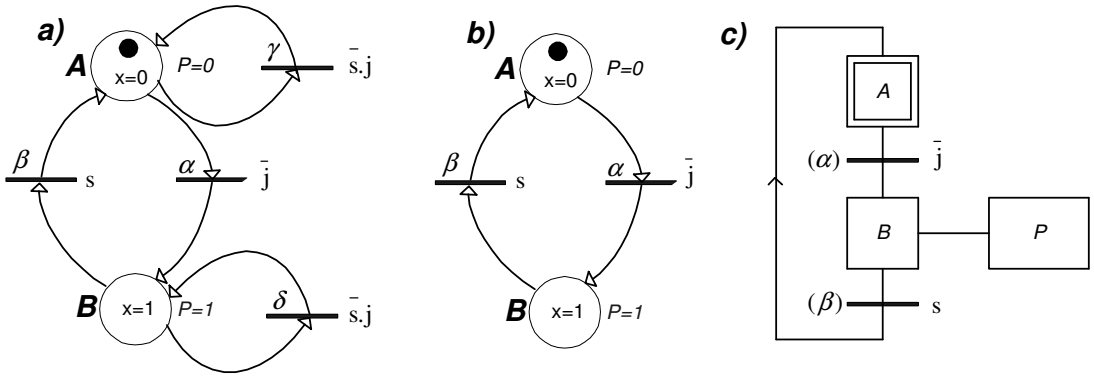


Fig. 4.29. Variante ale caietului de sarcini pentru automatul secvențial care reglează bipoziționalnivelul într-un rezervor.

Se observă că automatul are structura tipică a unui automat cu neschimbarea stării. *Analiza structurala* a rețelei arată că conflictele structurale care apar în pozițiile ***A*** şi ***B*** nu sunt şi efective datorită evenimentelor care interpretează tranzițiile γ şi δ. Din această cauză buclele care pornesc din pozițiile ***A*** şi ***B*** pot fi înlăturate rezultând rețeaua Petri din Fig. 4.29 b). Această rețea fiind *conformă* are grafcetul echivalent din Fig. 4.29 c).

Sinteza automatului se face pe baza rețelei din Fig. 4.29 b). La început se *codifică* pozițiile. Având numai două poziții variabila de stare poate fi codificată cu ajutorul unui singur bit *x*. Deoarece în A pompa este oprită se ia valoarea *x=0*. Din acelaşi motiv se atribuie lui ***B*** valoarea *x=1. Analizând comportamental* rețeaua se poate completa tabelul de adevăr din Fig. 4.30. În primele două linii ale tabelului evenimentele s şi j nu au apărut[138]. Această situație coincide zonei din Fig. 4.28 în care nivelul este mai mic decât valoarea minimă. Dacă poziția A a automatului este activă este validată tranziția α şi deoarece evenimentul *j* nu a apărut avem relația (4.47).

$$\overline{j}=1 \qquad (4.47)$$

Expresia logică care interpretează tranziția α are valoarea 1 şi tranziția se declanşează deoarece este validată. Poziția ***A*** devine inactivă iar poziția ***B*** devine activă. Deci $x^{t+\Delta}=1$ şi se completează în mod corespunzător ultima coloană a tabelului de adevăr. În mod asemănător se completează toată coloana a patra a tabelului.

[138] Variabilele logice corespunzătoare evenimentelor au valorile s=*0* şi *j=0*.

s^t	j^t	x^t	$x^{t+\Delta}$
0	0	0	**1**
0	0	1	**1**
0	1	0	**0**
0	1	1	**1**
1	0	0	**-**
1	0	1	**-**
1	1	0	**0**
1	1	1	**0**

Fig. 4.30 Tabelul de adevăr al automatului pentru reglarea bipoziţională a nivelului în rezervor.

O problemă apare în liniile cinci şi şase ale tabelului de adevăr. Varianta în care *s=1* şi *j=0* nu este posibilă din punct de vedere fizic. Nivelul nu poate fi în acelaşi timp mai mare decât valoarea maximă şi mai mic decât valoarea minimă. Din acest motiv, pentru aceste linii în coloana a patra a tabelului de adevăr nu s-a trecut nici o valoare a variabilei de stare $x^{t+\Delta}$.

$x^{t+\Delta}$

x^t \ $s^t j^t$	00	01	11	10
0	1	0	0	-
1	1	1	0	-

Fig. 4.31 Diagrama Karnaugh a automatului pentru reglarea bipoziţională a nivelului.

Valorile din tabelul de adevăr Fig. 4.30 se trec în diagrama Karnaugh Fig. 4.31 şi se fac două grupări. Expresia logică rezultantă este dată de

$$x^{t+\Delta} = \overline{j^t} + \overline{s^t}.x^t$$

Considerând relaţia fizică a releului (4.37) şi relaţia de echivalenţă (4.38) se pot elimina variabilele *x, Δ* şi *t* rezultând relaţia (4.48).

$$K = \overline{j} + \overline{s}.k \qquad (4.48)$$

Implementarea cu contacte şi relee a aceste relații este prezentată în schema desfăşurată electrică din Fig. 1.6, circuitele 10 şi 20.

4.13. Automate elementare cu circuite integrate

Ca şi sistemele logice combinaționale, sistemele discrete logice simple, bistabile, pot fi implementate software sau cu circuite integrate logice. Sistemele bistabile implementate cu circuite electronice integrate poartă numele de *circuite bistabile* şi au proprietăți cu totul remarcabile care au condus la folosirea lor intensă nu numai în domeniul sistemelor de conducere dar şi la sistemele de comunicație sau la sistemele de calcul. Bistabilele pot fi asincrone sau sincrone.

4.13.1. Bistabilul RS.

În paragraful 4.12.5 am sintetizat relația logică (4.43) corespunzătoare rețelei bistabile cu prioritate la pornire din Fig. 4.15 k). Pentru implementarea sa cu circuite integrate logice, la fel ca şi în cazul implementării cu contacte şi releu, trebuiesc adăugate relația fizică a circuitului integrat şi relația de echivalare a variabilelor.

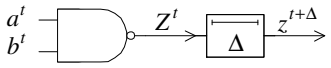


Fig. 4.32 Schema bloc a unui circuit integrat real.

Semnalul printr-un circuit electronic real nu se propagă instantaneu de la intrare la ieşire. Există o mică întârziere Δ care poate fi luată în considerare cu ajutorul unui bloc nou conectat la ieşirea circuitului ca în schema bloc din Fig. 4.32. Această caracteristică poate fi pusă în evidență, la fel ca şi în cazul releului, cu ajutorul relației (4.49).

$$z^{t+\Delta} = Z^t \tag{4.49}$$

Legătura dintre variabilele modelului sub formă de rețea Petri şi variabilele circuitului integrat este dată de relația (4.50). Eliminând variabilele *x, Δ* şi *t* din relațiile (4.43), (4.49) şi (4.50) rezultă (4.51).

$$x^t = z^t \tag{4.50}$$

$$Z = p + \bar{o}.z \tag{4.51}$$

Pentru a putea fi implementată cu ajutorul circuitelor ŞI – NU o relație scrisă în forma disjunctivă canonică trebuie[139] negată, trebuie să i se alice teorema De Morgan şi apoi trebuie negată din nou. Aplicând aceste operații relației (4.51) se obține (4.52) care poate fi implementată ca în Fig. 4.33.

$$Z = \overline{\overline{p}.\overline{(o.z)}} \qquad (4.52)$$

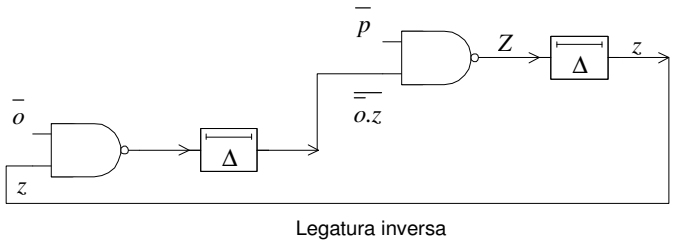


Fig. 4.33 Implementarea automatului cu prioritate la pornire cu circuite integrate ŞI – NU reale.

În Fig. 4.15 se fac următoarele simplificări:

- Toate blocurile de întârziere sunt aduse la ieşirea circuitului formând un singur bloc care nu mai este reprezentat în desen. Această simplificare face ca sistemul obținut să fie diferit de cel din Fig. 4.33. Aici cele două intrări au timpi de întârziere diferiți pe drumul lor de la intrarea la ieşirea circuitului. După simplificare întârzierile sunt acelea şi pentru cele două semnale de intrare.
- Se redesenează schema, aducând circuitul ŞI – NU de jos în dreptul celui de sus Fig. 4.34.
- Se consideră că apăsarea simultană pe butoanele de pornire şi de oprire este imposibilă.
- Semnalul de intrare, butonul de pornire *p,* este redenumit *S,* de la englezescul *Set.* Asemănător semnalul *o* devine *R,* de la *Reset*. Semnalul de ieşire *z* este redenumit *Q.*
- Se specifică pe desen o a doua ieşire, de la circuitul ŞI – NU de jos. În condițiile în care nu este posibil evenimentul unei apăsări simultane pe butonul *p* şi pe butonul *o*, cea de a doua ieşire este egală cu negata primei ieşiri.
- Tot în absența apariției simultane a celor două semnale de intrare se observă că *R* este egal cu negatul lui *S* şi invers.

[139] La fel ca şi în cazul sistemelor logice combinaționale.

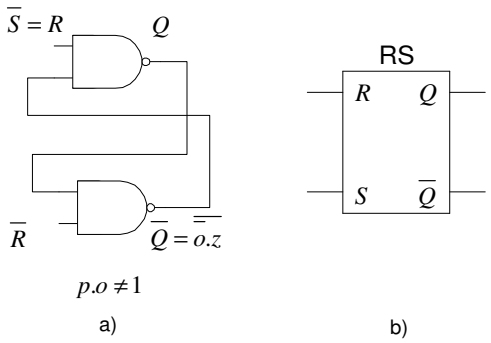


Fig. 4.34 Circuitul bistabil RS şi simbolul său.

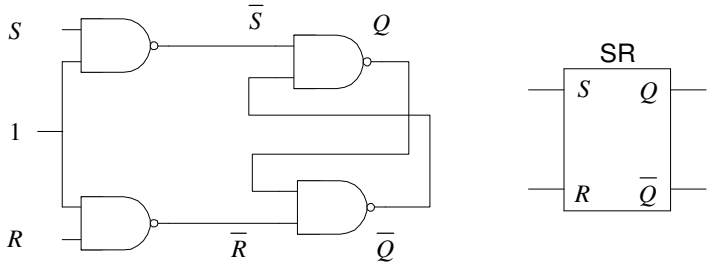


Fig. 4.35 Bistabilul SR şi simbolul său.

Rezultatul simplificărilor apare în Fig. 4.34.a). Cele doua circuite ŞI – NU sunt implementate într-un singur modul cu simbolul din Fig. 4.34 b). Circuitul bistabil are aceleaşi funcţiuni ca şi automatele elementare, memorează impulsurile de pornire (setare) sau oprire (resetare).

4.13.2. Bistabilele SR şi SRC. Sincronizarea.

Neajunsul că intrările bistabilului RS sunt negate se poate înlătura uşor cu ajutorul a două circuite ŞI – NU ca în Fig. 4.35 şi se obţine bistabilul SR.

Circuitele electronice sunt foarte rapide şi răspund la cele mai rapide perturbaţii. Comparativ cu acestea releul are o inerţie considerabilă care îl face imun la perturbaţiile de înaltă frecvenţă. Acelaşi efect se obţine la bistabile introducând *sincronizarea*.

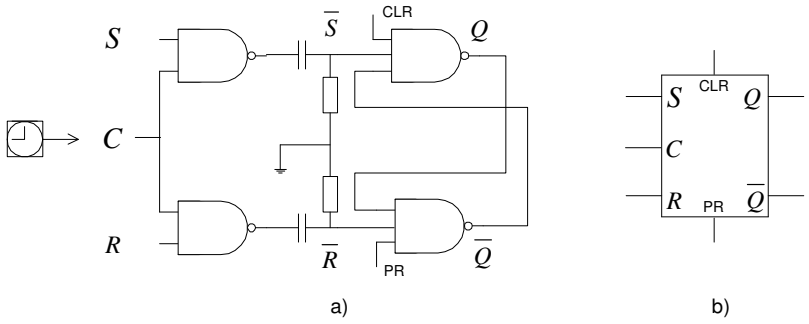


Fig. 4.36 Bistabilul SRC şi simbolul său.

În Fig. 4.36 a) s-a introdus o bornă suplimentară, de ceas[140], la care se trimit impulsuri periodice. Datorită condesatoarelor introduse în schemă comenzile de setare *S* şi resetare *R* sunt luate în considerare numai pe frontul crescător al impulsurilor de ceas. În felul acesta orice impuls parazit care apare între fronturile crescătoare ale impulsurilor de ceas sunt ignorate. Bornele CLEAR şi PRESET au acelaşi efect cu S negat şi, respectiv, R negat. Semnalele aplicate la aceste borne aduc ieşirea Q la nivelul 0, respectiv 1, într-un mod asemănător bistabilului RS.

4.13.3. Bistabilul SCR Master – Slave.

O măsură suplimentară de protecție la impulsuri perturbatoare care poate fi luată în cazul circuitelor integrate constă în ruperea legăturii electrice directe dintre intrare şi ieşire. Pentru aceasta se folosesc două bistabile SRC ca în Fig. 4.37. Primul bistabil funcționează pe frontul crescător al impulsului de ceas şi al doilea bistabil transmite semnalul la ieşire pe frontul descrescător al impulsului de ceas.

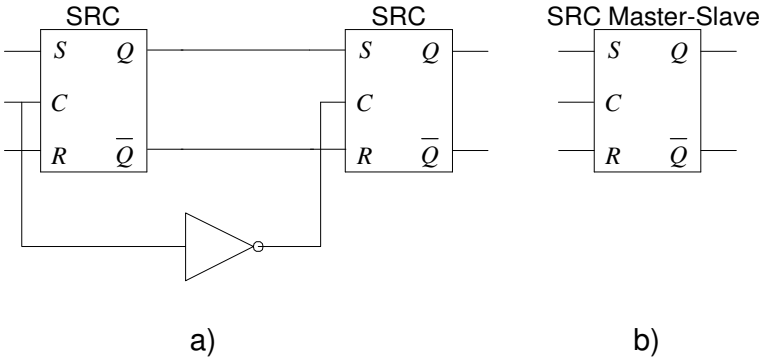


Fig. 4.37 Bistabilul SRC Master – Slave şi simbolul său.

[140] Clock (eng.)

4.13.4. Bistabilul JKC.

Dacă se introduc două legături inverse suplimentare la un bistabil SRC, aşa cum se vede în Fig. 4.38, se obține bistabilul JKC.

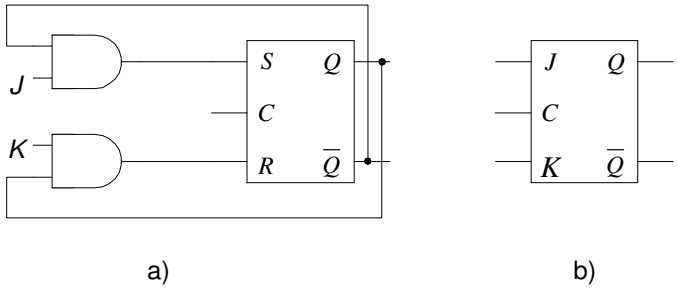


Fig. 4.38 Bistabilul JKC şi simbolul său.

Borna la care se aplică semnalul de oprire se numeşte acum J[141] iar borna la care se aplică semnalul de oprire se numeşte K[142]. Datorită legăturilor inverse trecerea circuitului dintr-o stare în alta, bascularea sa , se face acum mult mai rapid decât în cazul bistabilului SRC. Mai mult, la aplicarea simultană a semnalului de pornire şi oprire, *J=1* şi *K=1,* circuitul basculează, la fel ca şi în cazul automatului elementar cu contacte şi releu care are drept model rețeaua Petri din Fig. 4.15 i). Relația logică (4.36) se transformă cu noile notații în (4.53)

$$Q^{t+\Delta} = J^t.\overline{Q^t} + \overline{K^t}.Q^t \qquad (4.53)$$

4.13.5. Bistabilul D.

O conexiune ca în Fig. 4.39 transformă bistabilul JKC într+un bistabil cu o singură intrare *D*. Dacă se înlocuieşte în (4.53) *J* cu *D,* iar *K* cu negatul lui *D* se obține relația logică (4.54). Această relație este identică cu relația fizică a releului, deci bistabilul D poate realiza aceleaşi funcțiuni logice ca un releu.

[141] Jam (eng.)

[142] Keep (eng.)

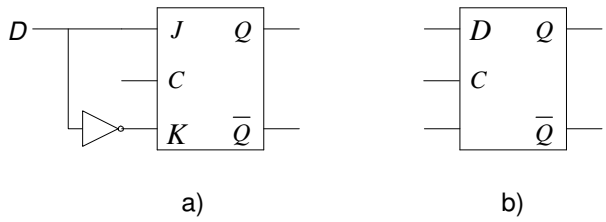


Fig. 4.39 Bistabilul D şi simbolul său.

$$Q^{t+\Delta} = D^t.\overline{Q^t} + \overline{\overline{D^t}}.Q^t = D^t \tag{4.54}$$

4.13.6. Bistabilul T.

O legătură simpla ca în Fig. 4.40 transformă bistabilul JKC în bistabilul T.

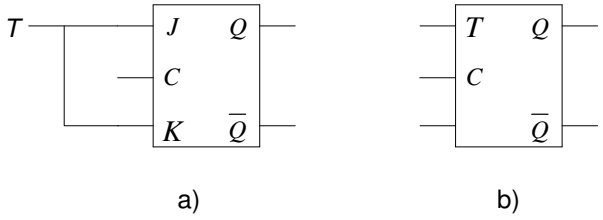


Fig. 4.40 Bistabilul T şi simbolul său.

Înlocuind în (4.54) noile valori ale lui J şi K se obţine relaţia logică (4.55) a bistabilului T. Dacă *T=0* ieşirea bistabilului rămâne neschimbată. Dacă *T=1* ieşirea basculează, adică dacă era 0 devine 1 şi dacă era 1 devine 0.

$$Q^{t+\Delta} = T^t.\overline{Q^t} + \overline{T^t}.Q^t = T^t \oplus Q^t \tag{4.55}$$

4.14. Proiectarea automatelor implementate cu contacte şi relee

Metoda de proiectare va fi prezentată pe baza exemplului automatului care porneşte un motor electric şi după o rotaţie să îl opreşte în poziţia iniţială.

4.14.1. Schema tehnologică şi schema bloc

Primul pas în proiectarea unui sistem automat constă în alegerea soluţiei de conducere automată[143]. În cazul de faţă se doreşte o soluţie simplă, care să nu controleze continuu poziţia axului motorului. Motorul trebuie să pornească la acţionarea butonului de pornire *p* şi să se oprească la acţionarea contactului *a* care indică terminarea unei rotaţii. Algoritmul de conducere prezentat ia în considerare numai evenimente discrete logice şi prin urmare sistemul automat va fi de tip *discret logic*.

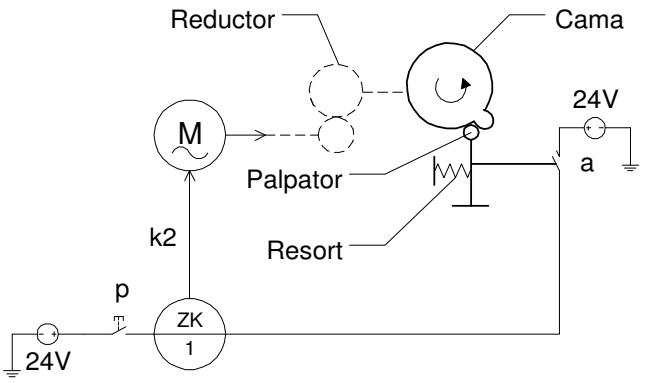


Fig. 4.41Schema tehnologica cu echipamentul de automatizare a sistemului automat discret logic pentru oprirea motorului electric după o rotaţie.

Automatistul împreună cu tehnologul[144] elaborează schema tehnologică cu echipamentul de automatizare din Fig. 4.41 şi schema bloc din Fig. 4.42. Modul în care se realizează aceasta nu poate fi algoritmizat. Soluţia aleasă depinde de foarte mulţi factori: banii disponibili pentru instalaţia de automatizare, aparatele de automatizare care pot fi procurate, mediul de lucru al sistemului, fiabilitatea impusă sistemului, calificarea necesară personalului de întreţinere, costul şi timpul acordat proiectului de automatizare, etc. In cazuri simple, ca acesta, se folosesc exemple asemănătoare[145].

[143] Alegerea soluţiei de automatizare.

[144] Adeseori sunt una şi aceeaşi persoană.

[145] Metoda de proiectare bazată pe exemple tipice.

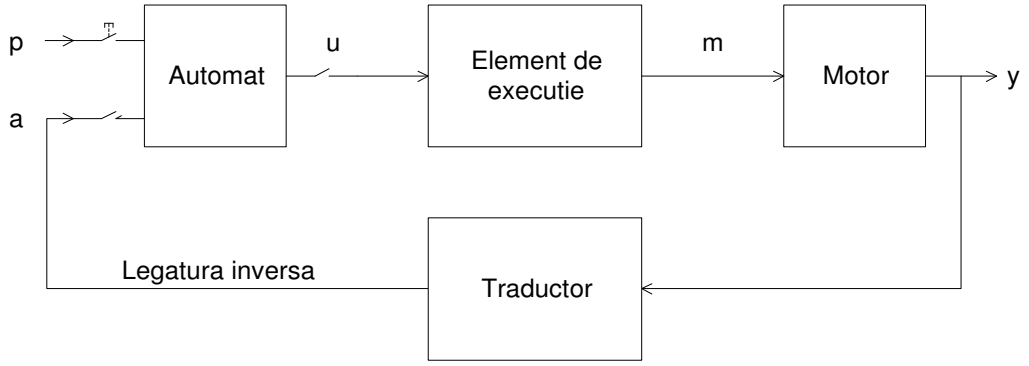


Fig. 4.42 Schema bloc a sistemului automat pentru oprirea motorului electric după o rotație.

Schema tehnologică cu echipamentul de automatizare din Fig. 4.41 prezintă instalația care urmează să fie condusă automat, motorul electric împreună cu traductorul de poziție şi stația de comandă cu numărul1.

Traductorul de poziție este format din reductorul de turație, cama, palpatorul şi contactul *a.* La terminarea unei rotații palpatorul acționează contactul *a* care se închide. Motorul se mai roteşte puțin şi datorită formei camei palpatorul deschide contactul *a*. În această poziție motorul trebuie să se oprească. Traductorul trebuie ales după caracteristicile tehnice necesare din cataloagele de specialitate sau trebuie proiectat şi construit.

Stația de comandă a poziției motorului are la intrare contactele *a* şi *p,* şi la ieşire contactul *k2*. Simbolul care o reprezintă indică faptul că stația de comandă se găseşte în alt loc decât motorul electric. Distanța dintre stație şi motor poate ajunge la câteva sute de metri. Din punct de vedere fizic stația de comandă este un dulap, cutie sau tablou care conține diferite componente şi aparate de automatizare. In mod curent stația de comandă poate conține:

- Alimentarea cu energie electrică a instalației tehnologice.
- Alimentarea instalației de automatizare.
- Automatul, automatul programabil, regulatoarele sau microcalculatorul de proces.
- Sistemul de semnalizare.
- Sistemul de protecție.
- Sistemul de comutare în diferite regimuri de funcționare.
- Sistemul de comunicare la distanță.

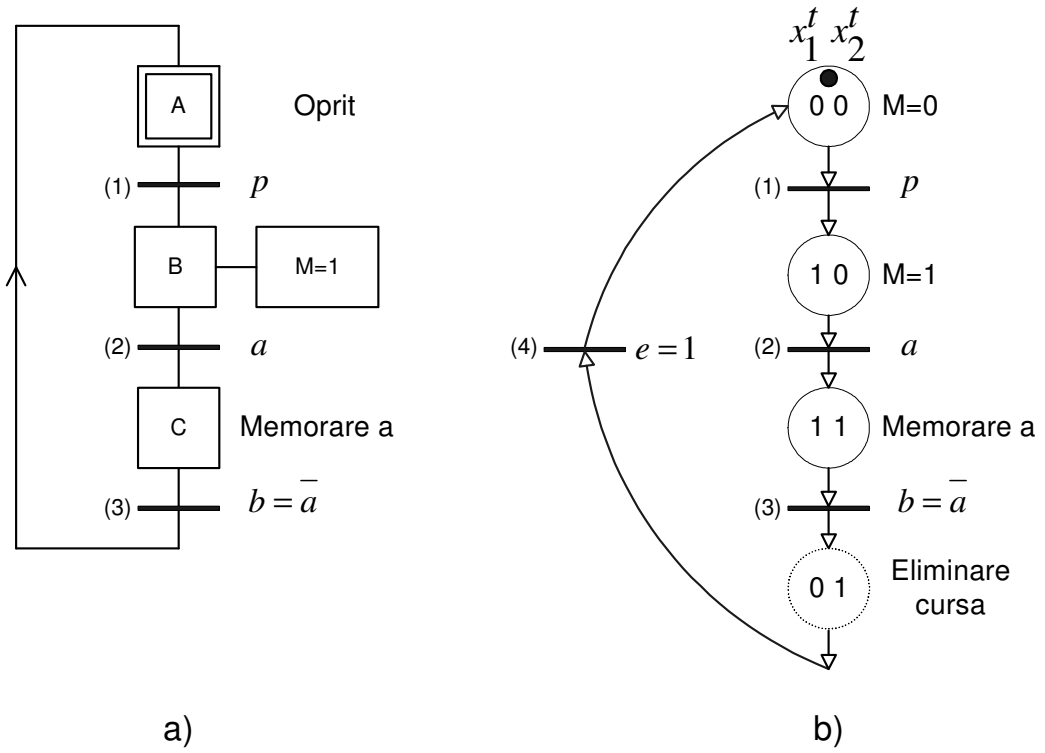


Fig. 4.43 Grafcetul şi rețeaua Petri echivalentă a automatului pentru oprirea unui motor după o rotație.

Stațiile de comandă trebuiesc proiectate şi construite din punct de vedere hardware (mecanic, electric şi electronic) şi software. În această lucrare se va pune accentul, în special, pe proiectarea sistemului de conducere.

Schema bloc a sistemului automat din Fig. 4.42 pune în evidență componentele principale ale sistemului de conducere automată. Comparând schema bloc cu schema tehnologică din Fig. 4.41 se constată că stația de comandă conține, în acest caz automatul şi elementul de execuție. Motoarele electrice de putere mare nu pot fi pornite oricum. Există un procedeu de pornire care poate fi implementat cu contacte şi relee[146] sau cu ajutorul unui aparat electronic numit demaror progresiv[147] şi care are la bază, în esență, un microcalculator specializat.

Elementul de execuție este în cazul de față un element de comutare de forță realizat cu ajutorul unui contactor[148]. Şi elementul de execuție trebuie proiectat[149] şi construit.

[146] Un automat de pornire stea-triunghi, de exemplu.

[147] Soft starter (eng.)

[148] Comutatorul alimentării motorului electric poate avea şi alte tipuri de realizări, de exemplu folosind tiristoare.

[149] Din punct de vedere hardware şi software, dacă este cazul.

Schema bloc prezintă semnalele de intrare şi ieşire din fiecare bloc. În cazul de față se observă că atât automatul, cât şi elementul de execuție, au la intrare şi la ieşire semnale de tip contact.

În continuare se va prezenta în special proiectarea automatului, elementului de execuție, a sistemului de semnalizare şi comutare în diferite regimuri de lucru.

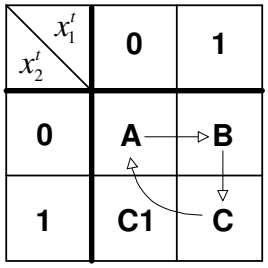


Fig. 4.44 Diagrama Karnaugh pentru codificarea adiacentă a pozițiilor rețelei Petri.

4.14.2. Caietul de sarcini al automatului

Caietul de sarcini. Algoritmul simplu de conducere al motorului este prezentat sub formă de grafcet în Fig. 4.43 a). Există trei etape şi trei tranziții. Etapa A este activă inițial şi prin convenție, dacă nu este specificat altfel, se consideră că toate acțiunile, exprimate prin expresii logice, iau în această etapă valoarea *0*, adică valoarea logică *fals.* Deci şi motorul electric este oprit în această etapă.

4.14.3. Analiza structurală şi comportamentală

Analiza structurală. Să considerăm că există o rețea Petri echivalentă grafcetului. Structura sa este identică, iar din punct de vedere grafic pătratele (etapele) se înlocuiesc cu cercuri (poziții). În etapa inițială A se pun o singură marcă. Deoarece fiecare din cele trei tranziții are un singur arc de intrare şi un singur arc de ieşire Rețeaua Petri este de tip maşină de stare. Graful asociat rețelei Petri este conex deoarece din orice nod[150] se poate ajunge în oricare alt nod, mai mult este tare conex deoarece această performanță se realizează prin intermediul unor arce orientate. Teorema lui Commoner spune că dacă graful este tare conex atunci rețeaua este *viabilă*. Dacă marcajul inițial are o singură marcă, cum este cazul aici, atunci rețeaua este viabilă şi *sigură*. Analizând structura rețelei se observă imediat că nu există conflicte structurale, deci rețeaua Petri este conformă şi echivalentă cu grafcetul. Toate concluziile pe care te obținem lucrând cu rețeaua Petri sunt valabile şi pentru grafcet.

[150] Nodurile grafului sunt pozițiile. Laturile grafului sunt arcele orientate care leagă nodurile atunci când se ignoră tranzițiile.

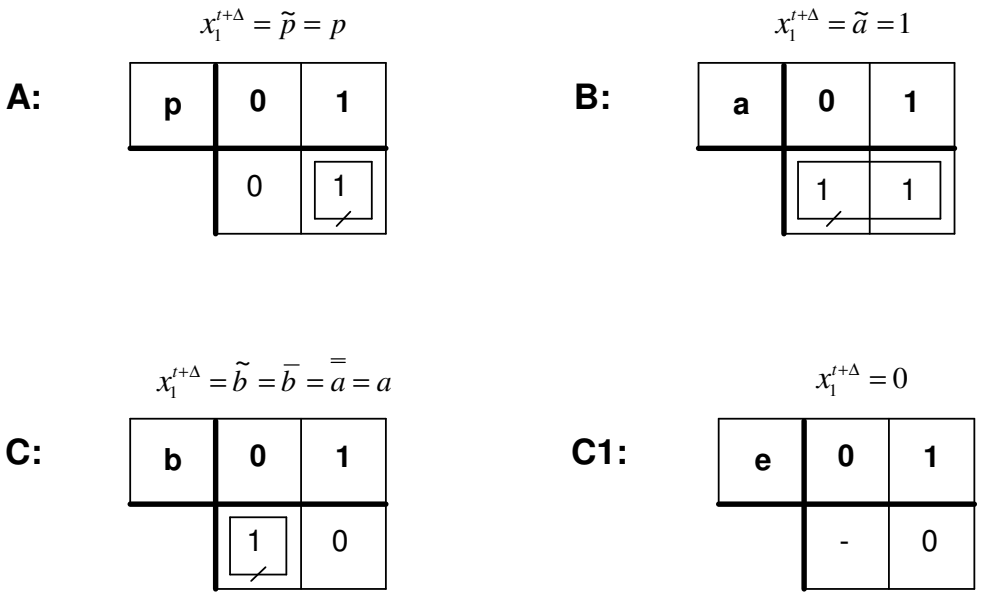


Fig. 4.45 Diagramele Karnaugh pentru stabilirea modului în care evoluează variabila de stare x_1 din fiecare poziție a rețelei din Fig. 4.43 a).

Analiza comportamentală. Tranziția (1), validată inițial, se declanşează când se apasă butonul de pornire şi *p=1*. Devine activă etapa B şi motorul porneşte. La terminarea unei rotații contactul traductorului este acționat de palpator şi *a=1*. Tranziția (2) se declanşează şi devine activă etapa C. Această etapă nu este interpretată de nici o acțiune nouă. Motorul se roteşte în continuare. Rostul acestei etape este să memoreze că a apărut evenimentul specific terminării unei rotații. Cama rotindu-se încet, la un moment dat palpatorul nu mai este împins şi contactul *a* nu mai este acționat. Expresia logică simplă care determină valoarea variabilei *b* va fi atunci egală cu unu (adevărat) şi tranziția (3) se declanşează. Etapa A redevine activă. Există deci un ciclu repetitiv care coincide cu algoritmul dorit pentru automatul sistemului.

Analiza comportamentală trebuie să ia în considerare şi alte situații de funcționare în afară de cea de bază. Ce se întâmplă dacă motorul se roteşte încet şi dorim să se oprească. Vedem că acest lucru nu este posibil. Conform caietului de sarcini motorul se roteşte întotdeauna până când se termină o rotație. Eventualele modificări ale caietului de sarcini se vor discuta o primă varianta a proiectului.

4.14.4. Sinteza automatului cu contacte şi relee.

Codificarea. Prima etapă a sintezei constă în codificarea binară a pozițiilor[151]. Având numai trei poziții A, B şi C numărul de biți necesari este egal cu doi. Într-adevăr, cu doi biți se pot obține $2^n=2^2=4$ coduri. Pentru a avea cât mai puține relee şi pentru a nu se modifica la

[151] Etapelor, deoarece grafcetul şi rețeaua Petri sunt echivalente.

fiecare tranziție decât un bit[152], se alege codul binar Gray. Deoarece capetele de tabele ale diagramelor Karnaugh sunt scrise în cod Gray putem folosi această diagramă şi la codificare. Pentru aceasta se trec denumirile pozițiilor rețelei în pătrate adiacente[153] ale diagramei Karnaugh cu două variabile. Din acest motiv această codificare se mai numeşte *adiacentă*. Există mai multe soluții de codificare. Una dintre ele se vede în Fig. 4.44. Deoarece pătratele diagramei în care se găsesc pozițiile A şi C ale rețelei nu sunt adiacente se introduce o poziție nouă C1 în rețea. Această poziție nu este interpretată de nici o acțiune. Rolul ei este de a asigura modificarea unui singur bit atunci când se trece de la poziția C la poziția inițială A.

Codurile obținute pentru cei doi biți de stare x_1 şi x_2 sunt prezentate în Fig. 4.43 b). Se observă că la trecerea de la o poziție la alta se modifică un singur bit. De exemplu, la declanşarea tranziției (2) poziția C devine activă iar poziția B inactivă. Bitul x_2 trece de la valoarea 0 la valoarea 1.

$x_1^{t+\Delta}$

x_2^t \ x_1^t	0	1
0	p	1
1	0	a

a)

$x_2^{t+\Delta}$

x_2^t \ x_1^t	0	1
0	0	a
1	0	1

b)

M^t

x_2^t \ x_1^t	0	1
0	0	1
1	-	1

c)

Fig. 4.46 Diagramele Karnaugh cu variabile înglobate ale variabilelor de stare şi de ieşire în cazul automatului pentru oprirea unui motor după o rotație.

Evoluția variabilelor de stare. Fiecare poziție este caracterizată de două variabile de stare: x_1 şi x_2. De exemplu, atunci când poziția B este activă[154] x_1=1 şi x_2=0.

Acestea sunt valorile biților la momentul actual (prezent) de timp *t*. În viitorul apropiat, la momentul de timp imediat următor *t+Δ,* poziția B poate să mai fie sau să nu mai fie activă. Dacă rămâne activă biții îşi păstrează valoarea. Atunci când tranziția (2) se declanşează biții îşi modifică, eventual, valorile. Aceste valori noi depind de apariția

[152] Dacă s-ar modifica teoretic simultan mai mulți biți s-ar pune problema practică ce se întâmplă dacă totuşi un bit se modifică mai rapid decât altul. Pentru a se evita rezolvarea problemei curselor dintre biți, care u7neori pot deveni critice, se adoptă măsura de precauție a modificării unui singur bit la declanşarea unei tranziții.

[153] Pătratele adiacente au o latură comună în sensul definiției diagramei Karnaugh care prevede că laturile opuse ale diagramei sunt lipite.

evenimentului *a* care interpretează tranziția (2). De exemplu, pentru stabilirea relației logice care determină valoarea bitului x_1 pentru poziția B se completează diagrama Karnaugh corespunzătoare acestei poziții din Fig. 4.45. Dacă evenimentul *a* nu apare, adică a=0, B rămâne activă şi x_1=1. Dacă evenimentul *a* apare, a=1, poziția C devine activă şi valoarea lui x_1 rămâne egală tot cu 1. Făcând gruparea corespunzătoare rezultă relația logică. Se procedează la fel pentru toate pozițiile şi pentru toți biții. Cu relațiile obținute se completează diagramele Karnaugh cu variabile înglobate din Fig. 4.46.

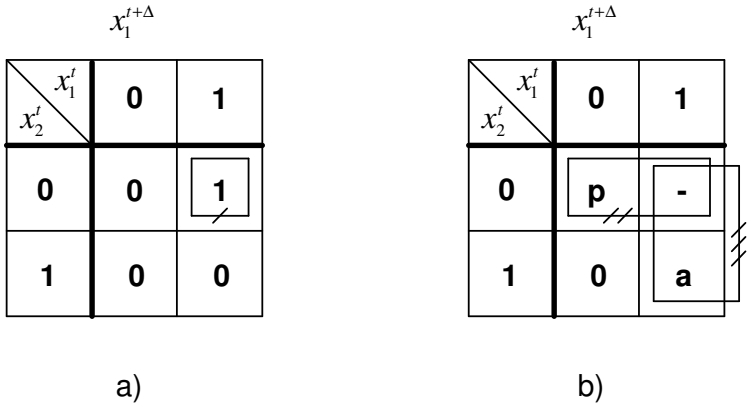


Fig. 4.47 Grupările pentru variabila de stare x_1 din Fig. 4.46 a).

Stabilirea relației logice pe baza diagramei Karnaugh cu variabile înglobate se face în doi paşi.

- Pasul 1. Variabilele înglobate se fac egale cu zero şi diagrama se prelucrează în mod clasic. De exemplu, pentru variabila de stare x_1 definită în Fig. 4.46 se obține diagrama din Fig. 4.47 a) cu o singură grupare.
- Pasul 2. În locul valorilor logice 1 se trec valori imposibile din punct de vedere tehnologic[155], marcate cu o liniuță şi se fac grupări de pătrate care au aceiaşi variabilă înglobată. Pentru aceiaşi variabilă x_1 se obține diagrama din Fig. 4.47 b) cu două grupări. Când se determină relația logică la fiecare grupare se ataşează şi variabila logică înglobată.

Pentru cele trei grupări din Fig. 4.47 a) şi b) rezultă relația logică (4.56) pentru x_1.

[154] Marca se găseşte în ea.

[155] Aceste situații nu există în realitate şi se poate considera o valoare logică 0 sau 1 astfel încât gruparea să fie cât mai mare şi deci simplificarea funcției logice să fie cât mai puternică.

$$\begin{aligned} x_1^{t+\Delta} &= \tilde{x}_1^t.\tilde{x}_2^t + p^t.\tilde{x}_1^t.\tilde{x}_2^t + a^t.\tilde{x}_1^t.\tilde{x}_2^t \\ x_1^{t+\Delta} &= x_1^t.\overline{x_2^t} + p^t.1.\overline{x_2^t} + a^t.x_1^t.1 \\ x_1^{t+\Delta} &= x_1^t.(\overline{x_2^t} + a^t) + p^t.\overline{x_2^t} \end{aligned} \qquad (4.56)$$

Diagramele Karnaugh din Fig. 4.46 b) şi c) permit calculul relaţiilor logice (4.57) şi (4.58)pentru x_2 şi M.

$$x_2^{t+\Delta} = x_1^t.(x_2^t + a^t) \qquad (4.57)$$

$$M^t = x_1^t \qquad (4.58)$$

4.14.5. Implementarea automatului cu contacte şi relee.

Alături de ecuaţiile de stare (4.56), (4.57) şi ieşire (4.58)se consideră relaţiile fizice ale releelor cu care implementăm cei doi biţi şi relaţiile de echivalenţă necesare.

$$\begin{aligned} k_1^{t+\Delta} &= K_1^t \\ k_2^{t+\Delta} &= K_2^t \\ x_1^t &= k_1^t \\ x_2^t &= k_2^t \end{aligned} \qquad (4.59)$$

Eliminând variabilele x_1, x_2, t şi Δ se obţin relaţiile logice (4.60) ale unui sistem logic combinaţionale format din două relee cu bobinele K_1 şi K_2 şi relaţia de ieşire.

$$\begin{aligned} K_1 &= k_1.(\overline{k}_2 + a) + p.\overline{k}_2 \\ K_2 &= k_1.(k_2 + a) \\ M &= k_1 \end{aligned} \qquad (4.60)$$

Schema electrică corespunzătoare relaţiilor logice (4.60) este prezentată în Fig. 4.48.

După implementare se face simularea automatului pentru a vedea dacă respectă caietul de sarcini din Fig. 4.43 a). La apăsarea butonului de pornire este acționat releul K1 din Fig. 4.48 care se automenține. În acelaşi timp porneşte motorul M care începe să se rotească. La terminarea unei rotații se acționează contactul *a* care provoacă atragerea releului K2 care se automenține. Motorul se roteşte în continuare. În momentul în care cama nu mai împinge palpatorul se deschide contactul *a* Ca urmare se realizează strict următoarea secvență de acțiuni: 1) releul K1 nu mai este acționat, 2) se opreşte motorul M şi releul K2 nu mai este acționat. Se ajunge în starea inițială şi se poate relua ciclul dacă se acționează butonul mde pornire *p.*

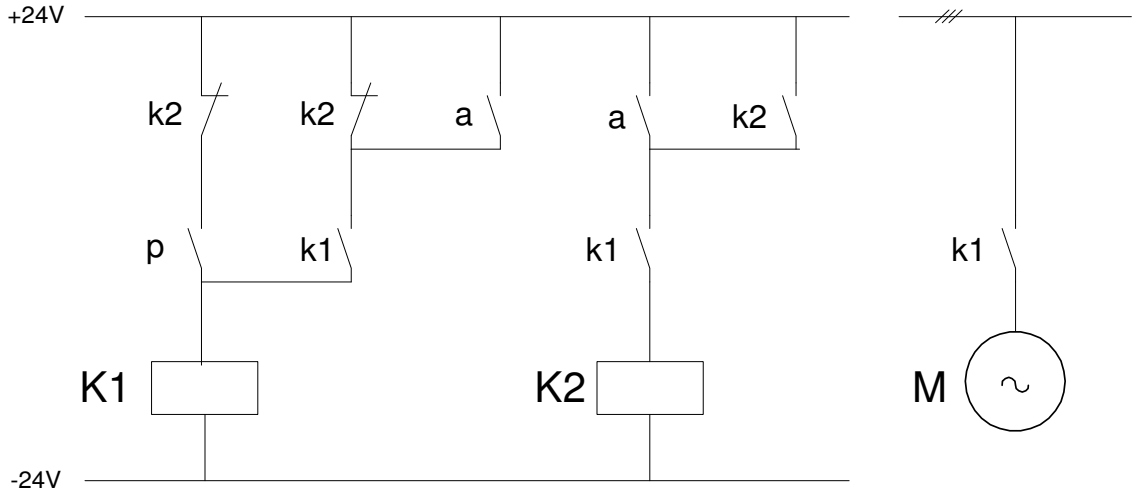


Fig. 4.48 Rezultatul sintezei şi implementării cu contacte şi relee a automatului pentru oprirea unui motor după o rotație.

Rezultatul sintezei şi implementării automatului din Fig. 4.48 nu este schema electrică desfăşurată finală. Pentru obținerea acesteia mai trebuiesc efectuate multe operații de proiectare. Să menționăm acum câteva dintre cele mai importante.

- Se introduce un sistem de alimentare a automatului.
- Se introduce un sistem de protecție a motorului, rețelei electrice de alimentare şi echipamentului de automatizare. Se foloseşte metoda exemplelor tip.
- Se introduce un sistem de semnalizare a principalelor stări, acțiuni şi evenimente care apar în funcționarea sistemului. Se foloseşte metoda exemplelor tip.
- Se introduce un sistem de deparazitare a perturbațiilor pe care le poate provoca în timpul funcționării automatul. Se foloseşte metoda exemplelor tip.
- Se dimensionează şi se aleg din catalog componentele automatului.
- Se dimensionează cutia în care se montează componentele automatului.
- Se întocmeşte documentația necesară pentru realizarea tehnologică a automatului.

4.15. Proiectarea automatelor implementate cu bistabile

Se foloseşte în continuare exemplul automatului pentru oprirea motorului după o rotaţie.

4.15.1. Sinteza şi implementarea automatului cu bistabile tip D

Codificarea. Deoarece bistabilele master – slave îşi schimba starea numai la momente exacte de timp, determinate de impulsurile de ceas, nu există restricţii la modul de codificare. Cu alte cuvinte se pot modifica mai mulţi biţi simultan la trecerea de la o poziţie la alta. Pentru codificarea din Fig. 4.43 b) nu mai este necesară introducerea unei poziţii suplimentare şi se obţine cazul din Fig. 4.49 a). În cazul acesta la trecerea de la poziţia cu codul 11 la poziţia cu codul 00 se modifică simultan ambii biţi.

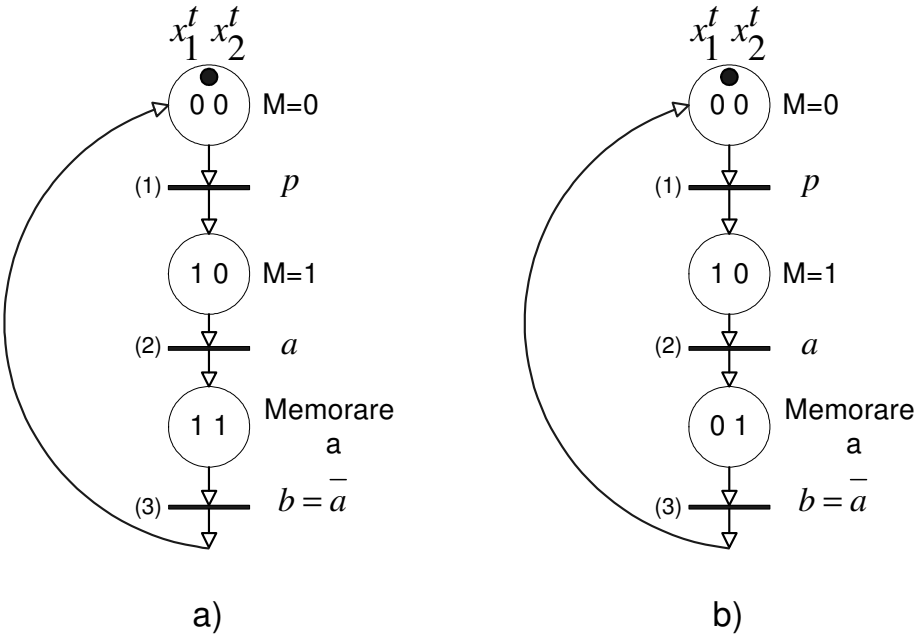


Fig. 4.49 Codificări neadiacente la sinteza automatului pentru oprirea unui motor după o rotaţie.

Pentru codificarea din Fig. 4.49 b) se modifică doi biţi când se declanşează tranziţia (2), iar în cazul codificării din Fig. 4.49 c), unde se folosesc trei biţi, se modifică doi dintre ei la declanşarea fiecărei tranziţii. Ultima codificare are avantajul că fiecare poziţie este caracterizată de un singur bit. În acest caz este mai uşor de înţeles funcţionarea şi de realizat sinteza şi implementarea. Preţul plătit pentru aceasta este un bistabil pentru fiecare poziţie.

Diagramele Karnaugh pentru biţii de stare şi variabila de ieşire ai reţelei din Fig. 4.49 a) sunt prezentate în Fig. 4.50. Relaţiile logice se stabilesc la fel ca în cazul sintezei cu contacte şi relee folosind metoda diagramelor Karnaugh cu variabile înglobate. Rezultă relaţiile (4.61). Să vedem problemele care apar în acest caz datorită modificării nesimultane a doi biţi în cazul implementării cu contacte şi relee.

$x_1^{t+\Delta}$

x_2^t \ x_1^t	0	1
0	p	1
1	0	a

a)

$x_2^{t+\Delta}$

x_2^t \ x_1^t	0	1
0	0	a
1	0	a

b)

M^t

x_2^t \ x_1^t	0	1
0	0	1
1	-	1

c)

Fig. 4.50 Diagramele Karnaugh pentru sinteza automatului corespunzător reţelei din Fig. 4.49 a).

$$\begin{aligned} x_1^{t+\Delta} &= x_1^t.(\overline{x_2^t} + a^t) + p^t.\overline{x_2^t} \\ x_2^{t+\Delta} &= a.x_1^t \\ M^t &= x_1^t \end{aligned} \qquad (4.61)$$

Adăugăm relaţiilor logice (4.61) relaţiile fizice ale releelor şi relaţiile de echivalenţă (4.59). După eliminarea variabilelor de stare x_1, x_2, a timpului t şi variaţiei de timp Δ rezultă sistemul de relaţii logice (4.62) care poate fi implementat cu contacte şi relee ca în Fig. 4.51.

$$\begin{aligned} K_1 &= k_1.(\bar{k}_2 + a) + p.\bar{k}_2 \\ K_2 &= k_1.a \\ M &= k_1 \end{aligned} \qquad (4.62)$$

Schema din Fig. 4.51 diferă de cea din Fig. 4.48 numai prin faptul că releul K2 nu are un contact de automenţinere. Schema este mai simplă, dar după activarea poziţiei caracterizate de codul 11 datorită acţionării contactului *a* poate apare următoarea situaţie. Ambele relee K1 şi K2 sunt acţionate prin circuite identice formate din contactele închise *a* şi *k1*. Când cama nu mai împinge palpatorul la terminarea unei rotaţii complete, Fig. 4.41, contactul *a* se deschide şi releele K1 şi K2 ar trebui să se dezactiveze *simultan*. Dacă din motive *fizice aleatorii* releul K2 se dezactivează mai încet decât K1, acesta din urmă se automenţine prin circuitul format din contactele închise k2 şi k1. Releul K1 rămânând acţionat motorul M merge în continuare

deşi ar trebui să se oprească deoarece s-a terminat o rotație. Din această cauză schema din Fig. 4.51 nu este acceptabilă în practică şi se foloseşte schema din Fig. 4.48 rezultată în urma unei codificări adiacente care asigură că nici o declanşare de tranziție nu va modifica simultan două variabile de stare.

Pentru implementarea cu bistabile master – slave tip D a relațiilor logice (4.61) obținute în urma unei codificări neadiacente se folosesc relațiile fizice ale bistabilelor D şi relațiile de echivalență a variabilelor de stare cu ieşirile bistabilelor (4.63).

$$\begin{aligned} Q_1^{t+\Delta} &= D_1^t \\ Q_2^{t+\Delta} &= D_2^t \\ x_1^t &= Q_1^t \\ x_2^t &= Q_2^t \end{aligned} \qquad (4.63)$$

După eliminarea variabilelor de stare x_1, x_2, a timpului t şi variației de timp Δ se obțin relațiile (4.64) pe baza cărora se elaborează schema electrică din Fig. 4.52.

$$\begin{aligned} D_1 &= Q_1.(\overline{Q_2} + a) + p.\overline{Q_2} \\ D_2 &= Q_1.a \\ M &= Q_1 \end{aligned} \qquad (4.64)$$

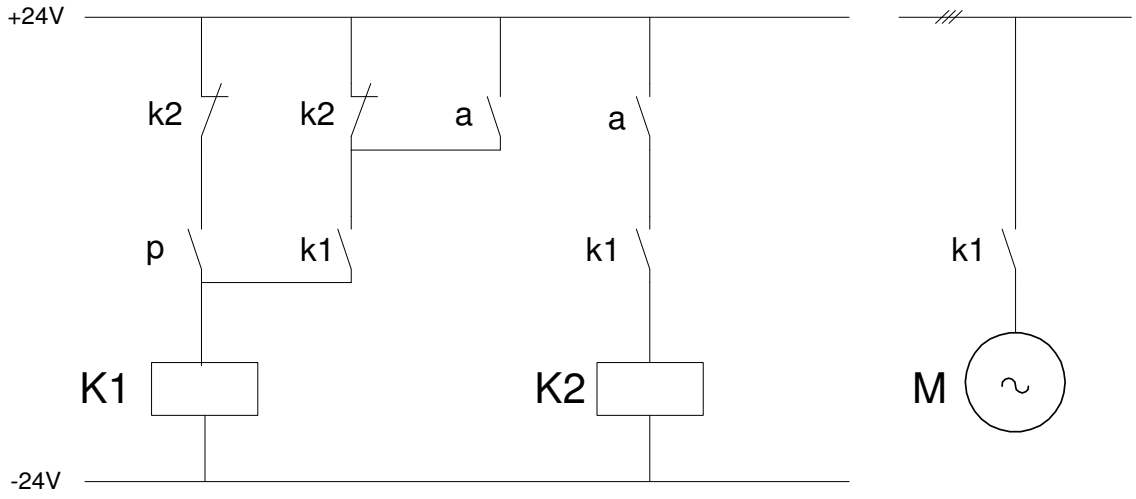


Fig. 4.51 Schema electrică în cazul codificării neadiacente din Fig. 4.49 a).

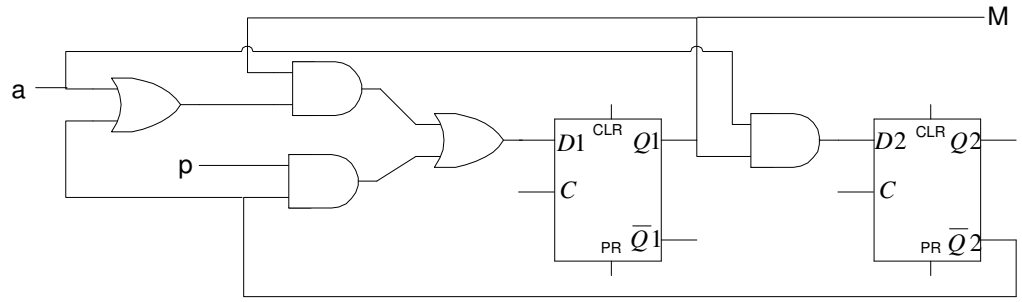


Fig. 4.52 Schema electrică a automatului pentru oprirea unui motor după o rotaţie în varianta implementată cu bistabile master – slave tip D.

4.15.2. Sinteza şi implementarea automatului cu bistabile JKC.

O altă variantă de codificare neadiacentă este prezentată în Fig. 4.49 b). De data aceasta se modifică doi biţi la declanşarea tranziţiei (2).

$x_1^{t+\Delta}$

x_2^t \ x_1^t	0	1
0	p	$\overline{a}$
1	0	0

a)

$x_2^{t+\Delta}$

x_2^t \ x_1^t	0	1
0	0	a
1	a	0

b)

M^t

x_2^t \ x_1^t	0	1
0	0	1
1	1	-

c)

Fig. 4.53 Diagramele Karnaugh pentru reţeaua din Fig. 4.49 b).

Evoluţia variabilelor de stare este prezentată în Fig. 4.53 a) şi b) iar variabila de ieşire în funcţie de variabilele de stare este determinată de diagrama Karnaugh din Fig. 4.53 c). Relaţiile logice corespunzătoare sunt (4.65).

$$\begin{aligned} x_1^{t+\Delta} &= p^t.\overline{x_2^t}.\overline{x_1^t} + \overline{a}.\overline{x_2^t}.x_1^t \\ x_2^{t+\Delta} &= a.x_1^t.\overline{x_2^t} + a.\overline{x_1^t}.x_2^t \\ M^t &= x_1^t + x_2^t \end{aligned} \qquad (4.65)$$

Din forma relațiilor fizice rezultă că se pot folosi la implementare bistabile JKC[156]. În (4.66) se prezintă relațiile fizice ale acestor bistabile şi relațiile de echivalență cu variabilele de stare.

$$\begin{aligned} Q_1^{t+\Delta} &= J_1^t.\overline{Q_1^t} + \overline{K_1^t}.Q_1^t \\ Q_2^{t+\Delta} &= J_2^t.\overline{Q_2^t} + \overline{K_2^t}.Q_2^t \\ x_1^t &= Q_1^t \\ x_2^t &= Q_2^t \end{aligned} \qquad (4.66)$$

Eliminând variabilele x_1, x_2, t şi Δ rezultă relațiile (4.67) care pot fi implementate ca în Fig. 4.54.

$$\begin{aligned} &J_1 = p.\overline{Q_2} \\ &\overline{K_1} = \overline{a}.\overline{Q_2}\ ,\ K_1 = a + Q_2 \\ &J_2 = a.Q_1 \\ &\overline{K_2} = a.\overline{Q_1}\ ,\ K_2 = \overline{a} + Q_1 \end{aligned} \qquad (4.67)$$

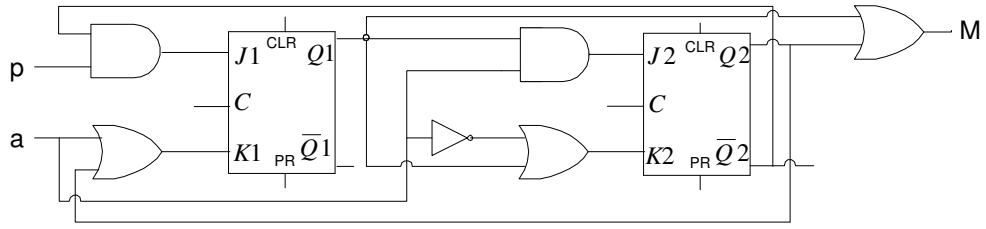


Fig. 4.54 Schema electrică a automatului pentru oprirea după o rotație în varianta implementată cu bistabile JKC master – slave.

4.15.3. Problema inițializării sistemului discret logic.

Problema inițializării, adică a stabilirii stării inițiale a sistemului[157], este adeseori tratată superficial deşi poate avea implicații mari asupra performanțelor sistemului.

[156] Bineînțeles că se pot folosi şi bistabile D, la fel ca în paragraful precedent.

[157] Stabilirea etapelor inițiale la grafcet sau a marcajului inițial la rețeaua Petri.

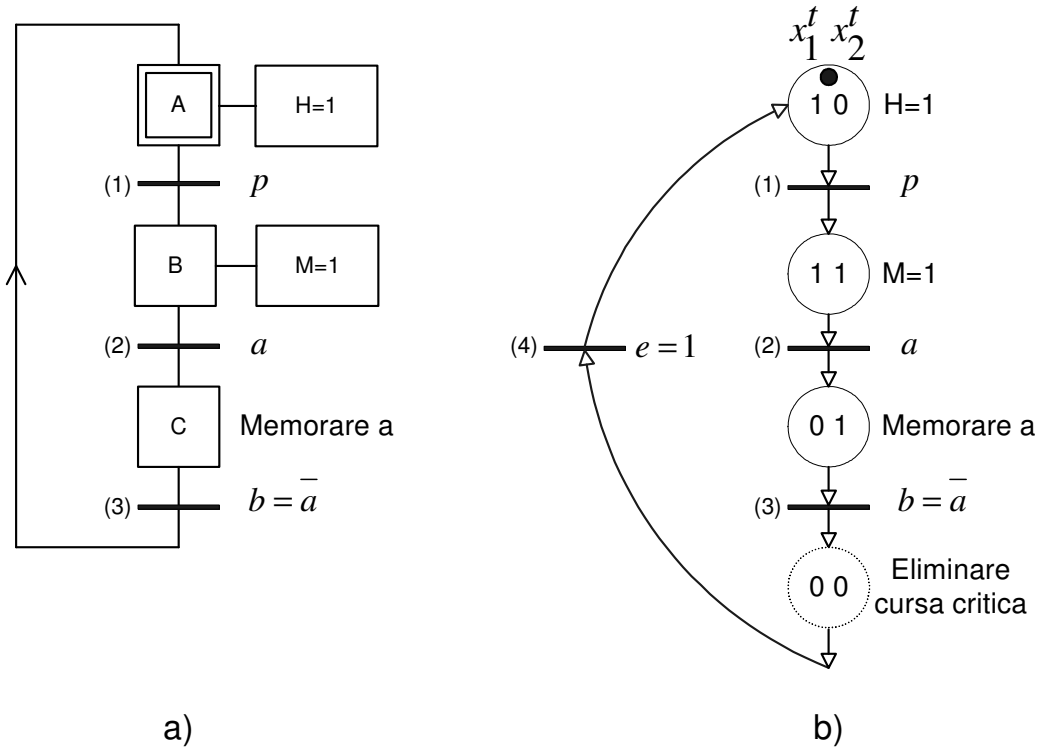


Fig. 4.55 Caietul de sarcini şi o codificare adiacentă pentru automatul care opreşte un motor după o rotaţie cu semnalizarea iniţializării.

La întocmirea caietului de sarcini din Fig. 4.43 pentru automatul care opreşte un motor electric după o rotaţie s-a considerat că etapa iniţială corespunde cu starea în care motorul este oprit, adică nealimentat. Dacă se acţionează butonul de pornire *p* motorul execută o rotaţie. Cu acest caiet de sarcini, în situaţia în care motorul se opreşte accidental, de exemplu la întreruperea alimentării cu energie electrică, la reluarea alimentării el ajunge în etapa A, Fig. 4.43 a), adică sistemul este iniţializat. La acţionarea butonului de pornire motorul nu mai execută însă o rotaţie completă! Acest lucru poate să fie foarte supărător în unele aplicaţii. Pentru eliminarea acestui inconvenient stabilim prin caietul de sarcini că în starea iniţială motorul trebuie să se găsească exact în situaţia din Fig. 4.41, adică într-o situaţie identică cu cea în care ajunge după terminarea unei rotaţii. Dacă se găseşte în starea iniţială, adică sistemul este iniţializat, lampa H semnalizează.

Pentru evitarea coincidenţii stării iniţiale cu starea oprit (nealimentat), la implementarea cu contacte şi releu, se codifică diferit de codul 00 poziţia iniţială a reţelei Petri Fig. 4.55 b) echivalente caietului de sarcini. O variantă de codificare adiacentă este cea prezentată în Fig. 4.56. Sinteza automatului se face la fel ca în paragraful 4.14.4. Se întocmesc diagramele Karnaugh din Fig. 4.57 pentru evoluţia variabilelor de stare şi pentru determinarea variabilelor de ieşire, se fac grupări şi se determină relaţiile logice (4.68) după metoda variabilelor înglobate, dacă se consideră relaţiile fizice şi de echivalenţă (4.59)

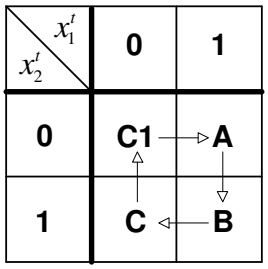


Fig. 4.56 Codificarea adiacentă a automatului cu inițializare.

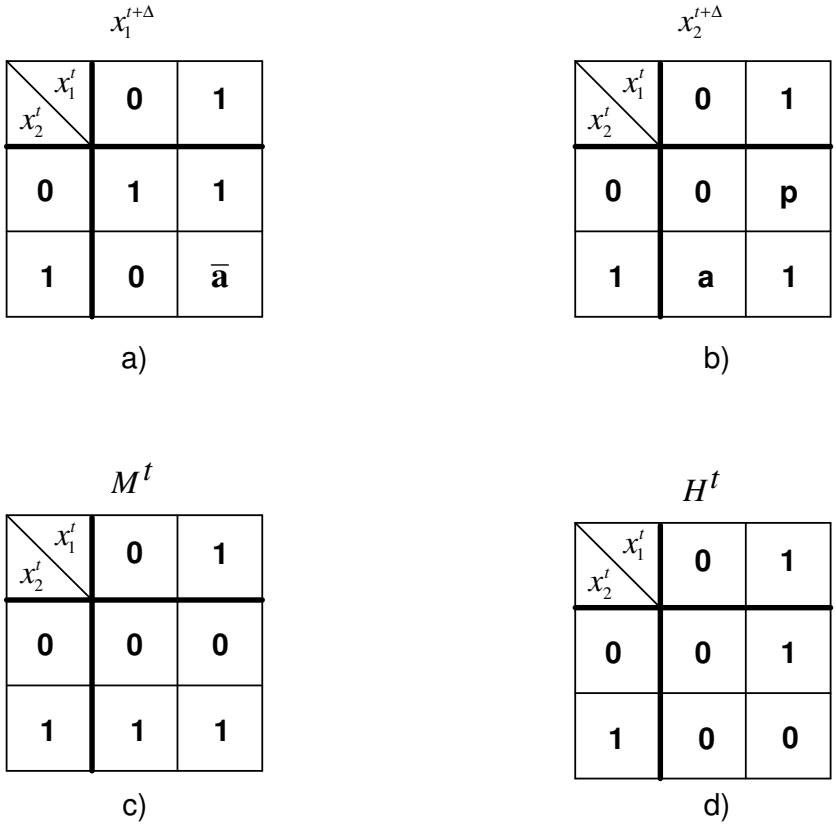


Fig. 4.57 Diagramele Karnaugh pentru evoluția variabilelor de stare şi pentru variabilele de ieşire în cazul automatului cu inițializare care opreşte un motor după o rotație.

Schema electrică corespunzătoare relațiilor (4.68) este prezentată în Fig. 4.58.

$$\begin{aligned} K_1 &= \overline{k}_2 + \overline{a}.k_1 \\ K_2 &= p.k_1 + (k_1 + a).k_2 \\ M &= k_2 \\ H &= k_1.\overline{k_2} \end{aligned} \qquad (4.68)$$

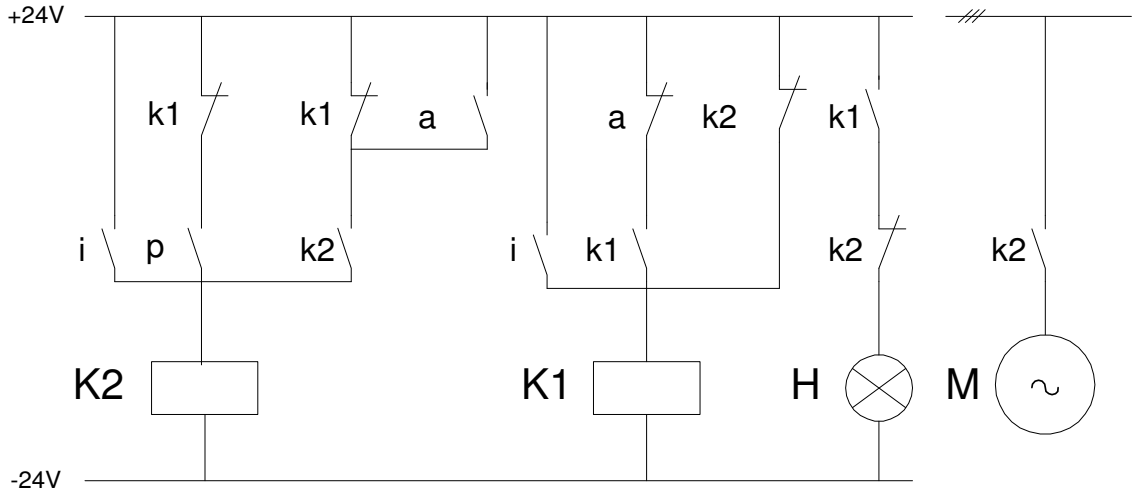


Fig. 4.58 Schema electrică a automatului cu inițializare pentru oprirea unui motor electric după o rotație.

Inițializarea sistemului se face prin apăsarea butonului i din Fig. 4.58. Se observă că în felul acesta sunt acționate ambele relee şi sistemul ajunge în poziția B din Fig. 4.55 b) codificată cu x1=1 şi x2=1. Motorul începe să se rotească, indiferent în ce poziție se află, execută o mişcare de rotație până când contactul a se deschide şi se închide la loc. După aceasta motorul se opreşte în poziția inițială A codificată cu x1=1 şi x2=0, iar lampa H se aprinde semnalizând inițializarea sistemului. După inițializare sistemul îşi poate relua ciclul normal de funcționare prin acționarea butonului de pornire.

La implementarea cu circuite bistabile inițializarea are aceiaşi importanță. Ea se face aplicând semnalele corespunzătoare pe bornele de ştergere CLR şi de aducere pe poziția unu logic PR ale bistabilelor, Fig. 4.52 şi Fig. 4.54. În acest caz codul 00 nu mai coincide cu situația în care bistabilele au fost puse în funcțiune prin alimentare cu energie electrică, aşa cum era cazul la implementarea cu contacte şi relee. Din această cauză inițializarea trebuie făcută întotdeauna, indiferent de semnificația poziției inițiale.

4.16. Proiectarea automatelor implementate cu APL

Pentru a prezenta câteva caracteristici a unei metode de proiectare pentru sistemele de conducere cu automate programabile trebuie, la început, să fac o delimitare a automatelor programabile de celelalte calculatoare care sunt folosite în cazul unui sistem de domotică. Aceasta este cu atât mai necesar cu cât exista automate programabile implementate pe PC, aşa numitele Soft-PLC.

Automatele programabile (AP) pot fi considerate microcalculatoare specializate care funcţionează în timp real, adică asigură o limită maximă pentru durata procesului de achiziţie, prelucrare şi redare a informaţiilor. Arhitectura tipică a unui AP este prezentată în Fig. 4.59. Se observă rolul important al intrărilor/ieşirilor logice şi analogice si al posibilităţilor de extensie.

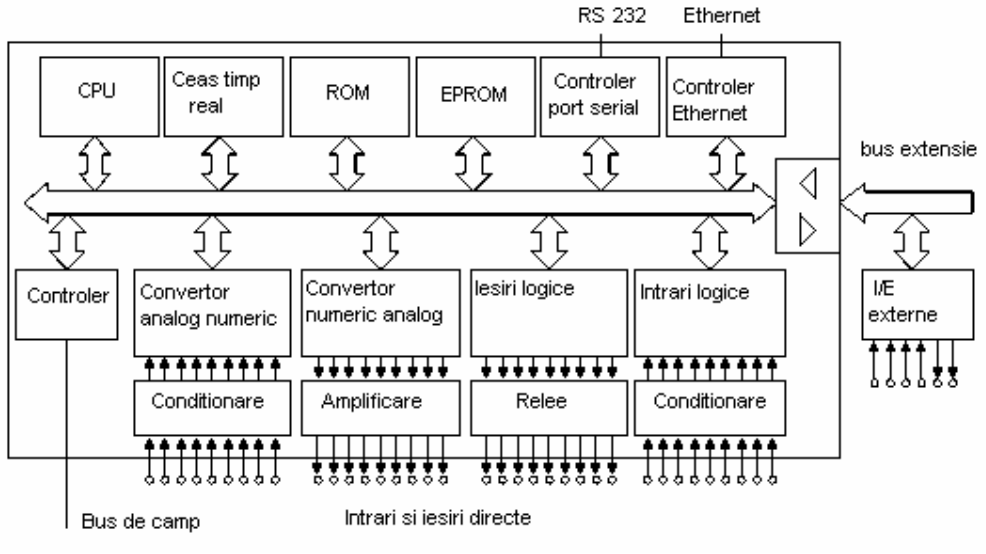


Fig. 4.59 Schema bloc a unui automat programabil

Intr-un sistem de domotică, Fig. 4.60, pot exista la nivelul de management şi nivelul de automatizare calculatoare care nu îndeplinesc condiţia de funcţionare în timp real, de exemplu diversele servere (web, baze de date, ftp, etc), gateway sau staţii clienţi. Sistemul lor de programare este diferit de cel pentru AP.

Deşi în Fig. 4.60 nu apar, aproximativ 20% dintre calculatoarele folosite în conducerea automată în timp real nu sunt de tip AP, principala deosebire constând în caracteristicile software care permit o programare mai elastică. Într-adevăr, schema de funcţionare pentru AP constă în scanarea intrărilor, executarea algoritmului de conducere, actualizarea ieşirilor şi realizarea operaţiilor de întreţinere. Proiectarea în această situaţie este concentrată asupra algoritmului de conducere deoarece ciclurile de intrare, ieşire şi întreţinere sunt ascunse. Celelalte tipuri de calculatoare folosite în conducerea automată, numite PAC (Programmable Automation Controller) oferă un acces mai profund la resursele hardware ale sistemului.

Şi din Fig. 4.60 se observă rolul important al intrărilor/ieşirilor şi comunicării prin magistrala (bus) de automatizare şi magistralele de câmp. Automatele programabile pot juca diferite roluri într-un sistem de domotică: interconectare directă I/E, Interconectare I/E prin intermediul magistralei de câmp sau gateway între magistrale.

Semnalele I/E logice şi analogice sunt prelucrate în mod diferit, după cum se observă în *Fig. 4.61* şi din această cauză şi metodele de proiectare sunt diferite. Eşantionarea cu o perioadă constantă este esențială atât pentru semnalele analogice cât şi pentru semnalele logice. In continuare mă voi referii numai la proiectarea sistemelor cu evenimente discrete care folosesc AP.

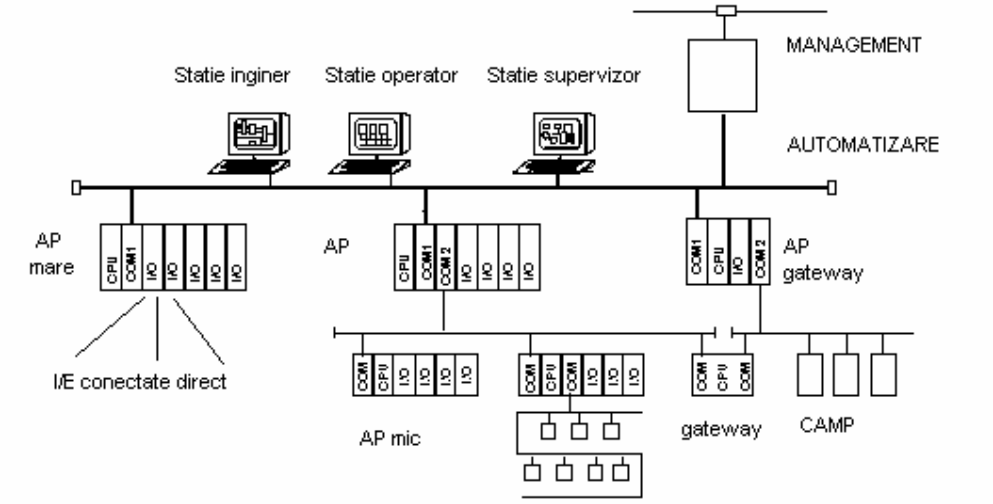


Fig. 4.60 Schema bloc a unui sistem de domotică.

Proiectarea unui sistem de conducere cu AP are cel puțin următoarele etape:

1. Identificarea procesului.
2. Stabilirea algoritmului de conducere şi a performanțelor necesare.
3. Programarea AP
4. Configurarea AP

Orice discuție privind proiectarea sistemelor de conducere cu automate programabile nu poate fi făcută în afara standardelor IEC 61131 şi IEC 61499. Standardul IEC 61131 are următoarele secțiuni:

- IEC 61131-1 Generalități
- IEC 61131-2 Testare
- IEC 61131-3 Programare şi tipuri de date
- IEC 61131-4 Ghidul utilizatorului
- IEC 61131-5 Comunicații
- IEC 61131-7 Conducerea Fuzyy

Evoluția standardelor referitoare la AP este prezentată în tabelul următor:

Anul	Standarde nationale	Standardul international
1977	DIN 40 719-6 (scheme bloc) GRAFCET (Franţa)	IEC 848
1983	DIN 19239 PLC programming Allen Bradley: limbaje de programare pentru AP	
1993	DIN EN 661131 Part 3	IEC 61131-3
1994	DIN EN 661131 Parts 1 and 2	
1995		IEC 61131-4
2005		IEC 61499 Conducere distribuita

O importanţă deosebită are standardul IEC 61131-3 care prevede următoarele modele pentru programarea AP:

1. LD (Ladder Diagram) – schemă desfăşurată cu contacte şi relee
2. FBD (Function Block Diagram) – schema bloc
3. IL (Instruction List) – program tip assembler
4. ST (Structured Text) – program tip Pascal
5. SFC (Sequential Function Charts) –grafcet

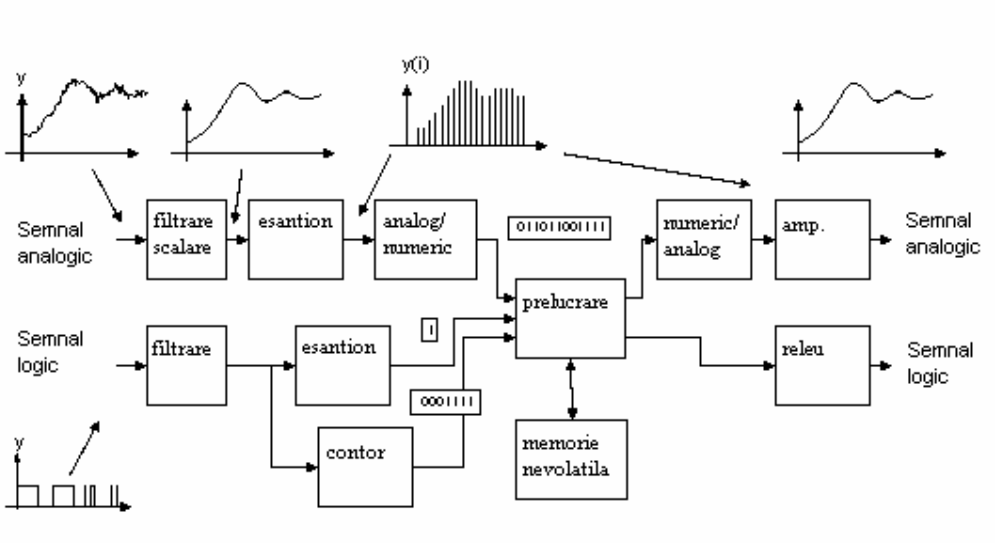


Fig. 4.61 Prelucrarea semnalelor logice şi analogice.

Pentru programarea sistemelor de conducere distribuite cu AP s-a elaborat standardul IEC 61499. În Fig. 4.62 se prezintă o schemă tehnologică simplă pentru un sistem de conducere distribuit format dintr-un PC, un AP, un regulator PID şi un robinet de reglare, toate interacţionând prin intermediul unei magistrale (bus). Un program sub formă de FBD conform IEC 61499 pentru acest sistem de conducere distribuit apare în aceiaşi figură. Între blocurile funcţionale există o legătură la nivelul fluxului de evenimente şi la nivelul fluxului de date. Fiecare bloc poate să aibă diferite intrări şi ieşiri cum ar fi referinţa SP şi ieşirea AUT pentru regulatorul PID. Blocul RESTART furnizează eveniment pentru execuţia periodică a celorlalte blocuri funcţionale.

În cadrul foarte general furnizat de aceste standarde vreau să prezint o modalitate de proiectare a sistemelor cu AP pe care am elaborat-o. Bineînțeles că metoda are multe limitări dar este mai simplă şi, lucru important, destul de algoritmizată, permițând şi inginerilor mai puțin experimentați proiectarea unor sisteme de conducere cu performanțe bune.

În primul rând, modelul pentru sistemul de conducere este o rețea Petri conformă şi nu un grafcet cum prevede IEC 61131-3. Grafcetul este mai general şi mai elastic pentru construcția de modele decât rețeaua Petri. În schimb modelul Petri este mai uşor de analizat şi tradus sub forma unui model matematic care să stea la baza elaborării programului pentru AP. Stabilirea relațiilor logice o fac cu ajutorul unor automate elementare cu prioritate la pornire, spre deosebire de soluția recomandată în general care foloseşte automate elementare cu prioritate la oprire. Soluția mea este mai aproape de spiritul grafcetului şi evită erori care pot apare la elaborarea unor modele care conțin paralelisme de tip conveyor. Metoda de proiectare va fi prezentată în continuare pe baza exemplului automatului care porneşte un motor electric şi după o rotație îl opreşte în poziția inițială, Fig. 4.41.

Primul pas în proiectarea unui sistem automat constă în alegerea soluției de conducere automată. În cazul de față se doreşte o soluție simplă, care să nu controleze continuu poziția axului motorului. Motorul trebuie să pornească la acționarea butonului de pornire *p* şi să se oprească la acționarea contactului *a* care indică terminarea unei rotații. Algoritmul de conducere prezentat ia în considerare numai evenimente discrete logice şi prin urmare sistemul automat va fi de tip *cu evenimente discrete*.

Automatistul împreună cu tehnologul elaborează schema tehnologică cu echipamentul de automatizare din Fig. 4.41. Modul în care se realizează aceasta nu poate fi algoritmizat. Soluția aleasă depinde de foarte mulți factori: banii disponibili pentru instalația de automatizare, aparatele de automatizare care pot fi procurate, mediul de lucru al sistemului, fiabilitatea impusă sistemului, calificarea necesară personalului de întreținere, costul şi timpul acordat proiectului de automatizare, etc. In cazuri simple, ca acesta, se folosesc exemple asemănătoare. *Schema tehnologică* cu echipamentul de automatizare prezintă instalația care urmează să fie condusă automat, motorul electric împreună cu traductorul de poziție şi stația de comandă cu numărul 1. *Traductorul de poziție* este format din reductorul de turație, cama, palpatorul şi contactul *a.* La terminarea unei rotații palpatorul acționează contactul *a* care se închide. Motorul se mai roteşte puțin şi datorită formei camei palpatorul deschide contactul *a*. În această poziție motorul trebuie să se oprească. Traductorul trebuie ales după caracteristicile tehnice necesare din cataloagele de specialitate sau trebuie proiectat şi construit. *Stația de comandă* a poziției motorului are la intrare contactele *a* şi *p,* şi la ieşire contactul *k2*. Simbolul care o reprezintă indică faptul că stația de comandă se găseşte în alt loc decât motorul electric. Distanța dintre stație şi motor poate ajunge la câteva sute de metri. Din punct de vedere fizic stația de comandă este un dulap, cutie sau tablou care conține diferite componente şi aparate de automatizare. In mod curent stația de comandă poate conține:

- Alimentarea cu energie electrică a instalației tehnologice.
- Alimentarea instalației de automatizare.
- Automatul, automatul programabil, regulatoarele sau microcalculatorul de proces.

- Sistemul de semnalizare.
- Sistemul de protecţie.
- Sistemul de comutare în diferite regimuri de funcţionare.
- Sistemul de comunicare la distanţă.

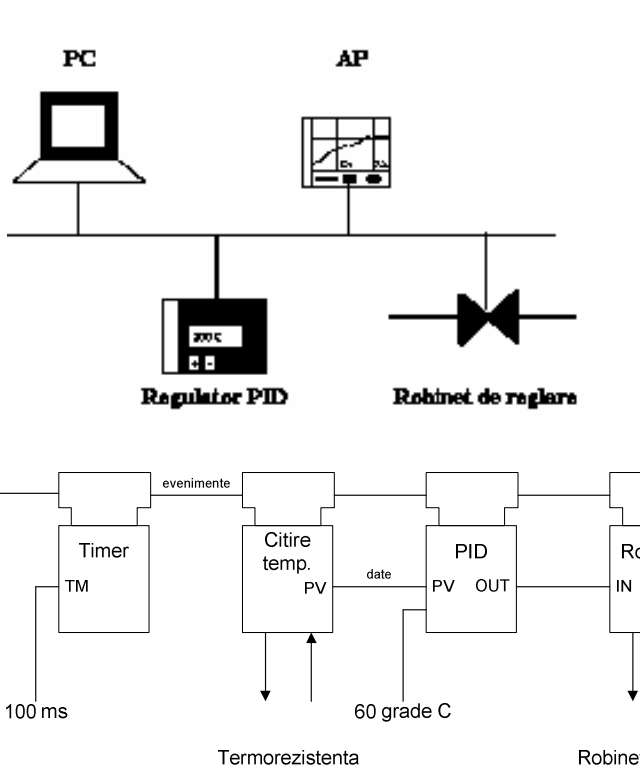


Fig. 4.62 Un sistem de conducere distribuit şi programul corespunzător tip FBD

Staţiile de comandă trebuiesc proiectate şi construite din punct de vedere hardware (mecanic, electric şi electronic) şi software. În această lucrare se va pune accentul, în special, pe proiectarea sistemului de conducere.

În continuare se va prezenta în special proiectarea automatului, elementului de execuţie, a sistemului de semnalizare şi comutare în diferite regimuri de lucru.

Caietul de sarcini. Algoritmul simplu de conducere al motorului este prezentat sub formă de grafcet în Fig. 4.63 a). Există trei etape şi trei tranziţii. Etapa A este activă iniţial şi prin convenţie, dacă nu este specificat altfel, se consideră că toate acţiunile, exprimate prin expresii logice, iau în această etapă valoarea *0*, adică valoarea logică *fals.* Deci şi motorul electric este oprit în această etapă.

În continuare se prezintă mai multe metode de proiectare a automatelor cu evenimente discrete care sunt implementate cu automate programabile logice şi se foloseşte acelaşi exemplu de automat pentru oprirea unui moto după o rotaţie.

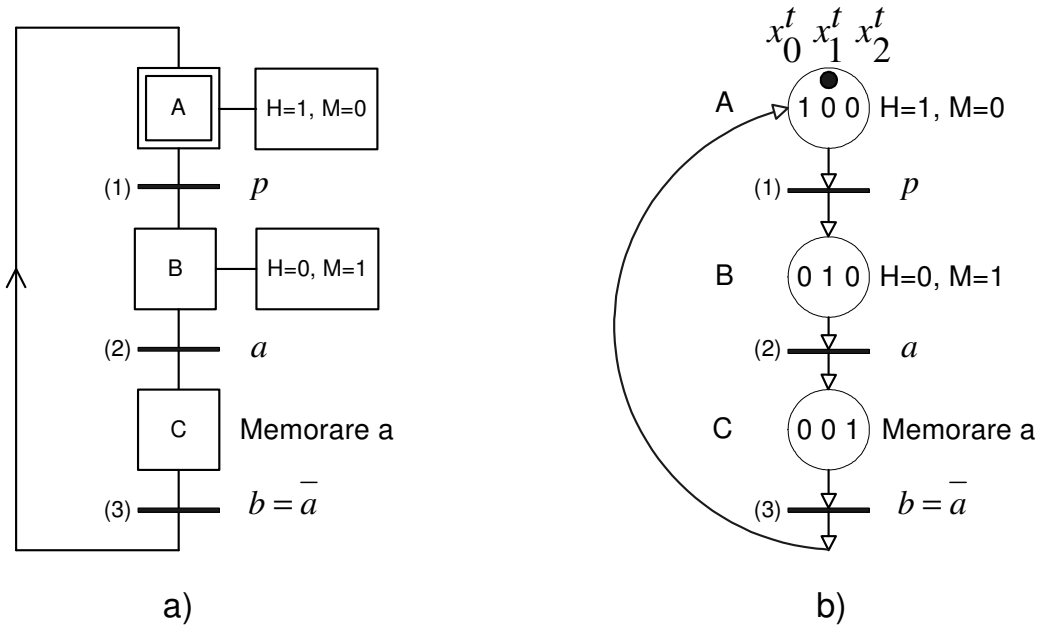


Fig. 4.63 Codificarea distribuită (1 din n) a caietului de sarcini cu inițializare pentru automatul care opreşte un motor după o rotație.

4.16.1. Metoda Grafcet

Caietul de sarcini Grafcet din Fig. 4.63 a) poate fi implementat direct la unele automate programabile folosind o interfață grafică. Metoda este foarte utilă în cazul automatelor complexe.

4.16.2. Metoda listei de instrucțiuni

Se construieşte şi se foloseşte rețeaua Petri ***conformă*** care descrie funcționarea automatului. Pentru exemplul considerat aceasta este prezentată în Fig. 4.63 b). Metoda de proiectare are următoarele etape:

Etapa 1. Codificarea locațiilor cu ajutorul codului distribuit (1 din n). Rezultatul apare în Fig. 4.63 b). Se observă că fiecărei locații îi corespunde un singur bit egal cu unu. Metoda de proiectare se bazează pe observația că acest bit poate fi implementat cu ajutorul unui automat elementar cu prioritate la pornire. Condiția de setare coincide cu condiția de activare a locației iar condiția de resetare coincide cu condiția de dezactivare a locației.

Etapa 2. Calculul condițiilor de setare şi resetare a biților corespunzători fiecărei etape. Setarea are loc dacă toate tranzițiile *precedente* locației se declanşează, iar resetarea se produce atunci când toate tranzițiile *posterioare* locației se declanşează. Declanşarea unei tranziții apare dacă este *validată* **şi** *evenimentul* care o interpretează a apărut, adică are valoarea logică unu. O tranzitie este validată atunci când toate locatiile precedente sunt active.

Exemplul studiat este mai simplu pentru că fiecare locație are câte o singură tranziție precedentă şi o singură tranziție posterioară

$$n=0 \quad s_0 = x_2 \bar{a} \quad r_0 = x_0 p \tag{4.69}$$

$$n=1 \quad s_1 = x_0 p \quad r_1 = x_1 a \tag{4.70}$$

$$n=2 \quad s_2 = x_1 a \quad r_2 = x_2 \bar{a} \tag{4.71}$$

Etapa 3. Determinarea relațiilor logice pentru locațiile active inițial cu ajutorul următoarei formule care provine din expresia automatului elementar cu prioritate la pornire:

$$x_n^{t+\Delta} = s_n^t + \overline{r_n^t} x_n^t + i \tag{4.72}$$

Se calculează simplificările posibile.

În exemplul nostru este activă inițial locația A caracterizată de bitul cu n=0. Înlocuind condițiile (4.69) în (4.72) se obține:

$$x_0^{t+\Delta} = x_2^t \bar{a} + \overline{(x_0^t p)} x_0^t + i = x_2^t \bar{a} + (\overline{x_0^t} + \overline{p^t}) x_0^t + i = x_2^t \bar{a} + \overline{p^t} x_0^t + i \tag{4.73}$$

Etapa 4. Determinarea relațiilor logice pentru locațiile inactive la momentul de timp inițial. Se foloseşte tot o formulă derivată din relația logică a automatului elementar cu prioritate la pornire:

$$x_n^{t+\Delta} = (s_n^t + \overline{r_n^t} x_n^t)\bar{i} \tag{4.74}$$

Pentru automatul din Fig. 4.63 b) locațiile B şi C nu sunt active inițial şi aplicând formula precedentă pentru condițiile (4.70) şi (4.71) se obține:

$$x_1^{t+\Delta} = (x_0^t p^t + \bar{a} x_1^t)\bar{i} \tag{4.75}$$

$$x_2^{t+\Delta} = (x_1^t + x_2^t) a^t \bar{i} \tag{4.76}$$

Etapa 5. Determinarea relațiilor logice dintre stările x şi ieşirile y ale automatului la momentul de timp t. Aceste relații rezultă din rețeaua Petri care specifică pentru fiecare locație, caracterizată prin anumite valori ale variabilelor de stare x, care valorile variabilelor de ieşire y. De exemplu, pentru rețeaua Petri din Fig. 4.63 b) variabila de ieşire y_m care

comandă motorul M are valoarea 1 numai pentru locațiile B şi C. Diagrama Karnaugh corespunzătoare şi relația logică stabilită[158] sunt prezentate în Fig. 4.64

$$y_m^t = x_1^t + x_2^t$$

x_2^t \ $x_0^t x_1^t$	00	01	11	10
0	0	1	-	0
1	1	-	-	-

Fig. 4.64 Determinarea relației logice pentru variabila de ieşire y_m care comandă motorul

$$y_h^t = x_0^t$$

x_2^t \ $x_0^t x_1^t$	00	01	11	10
0	0	0	-	1
1	0	-	-	-

Fig. 4.65 Determinarea relației logice pentru variabila de ieşire y_h care comandă lampa de semnalizare

Din aceiaşi rețea Petri rezultă că lampa H semnalizează numai în locația A şi deci numai atunci variabila de ieşire corespunzătoare y_h are valoarea 1. Relația logică corespunzătoare determinată cu ajutorul diagramei Karnaugh este prezentată în Fig. 4.65.

Etapa 6. Elaborarea tabelului de configurare. Se stabileşte o corespondență între variabilele relațiilor logice şi denumirile elementelor componente ale automatului

[158] Există tentația ca în cazul unor relații logice simple, cum este cazul exemplului prezentat, să se deducă direct relația logică fără ajutorul diagramei Karnaugh. Este o metodă greşită pentru că permite ignorarea anumitor situații care sunt puse însă în evidență de către diagrama Karnaugh. În exemplul prezentat acesta este cazul cu situația în care toate variabilele de stare sunt egale cu zero, situație care un apare în rețeaua Petri. Diagrama Karnaugh ne-a silit să precizăm valorile de ieşire în acest caz. Am hotărât, de exemplu, că în acest caz lampa de semnalizare H nu este aprinsă.

programabil: intrări, ieşiri, memorii (relee), timere (relee de timp), etc. Pentru exemplul considerat acestea apar în Tab. 4.2.

Operațiunile logice de bază pe care le realizează un automat programabil sunt prezentate pe scurt în continuare folosind schema bloc din Fig. 4.66.

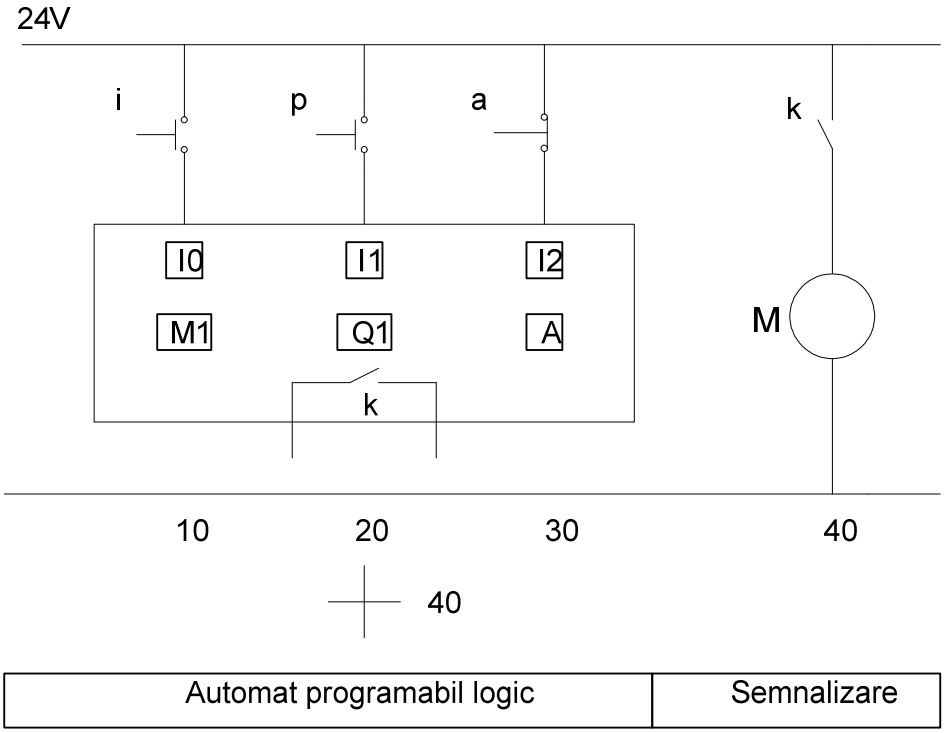


Fig. 4.66 Schema bloc a unui automat programabil logic

Un automat programabil logic are mai multe celule de memorie specializate, fiecare dintre ele caracterizate printr-o adresă[159]. Câteva tipuri de celule de memorie mai importante sunt următoarele:

- Intrările, de exemplu I1, I2 şi I3. La aceste intrări se conectează dispozitive care furnizează +24V pentru unu logic şi zero volți pentru unu logic. La intrare I3 este conectat un buton de oprire. El este normal închis , astfel în cât ruperea circuitului să echivaleze cu comanda oprit.
- Ieşirile. În Fig. 4.66 apare numai ieşirea Q1. Valoarea unu logic în celula de ieşire acționează un releu cu contactul k.
- Memoriile, În aceste celule, de exemplu M1, se memorează rezultatele intermediare.
- Acumulatorul A. Există un singur acumulator în care se realizează toate operațiile logice. Un operand trebuie să se găsească întotdeauna în acumulator. Rezultatul operației logice apare tot în acumulator.

[159] Adresele sunt numerotate începând cu zero în standardul IEC 61131-3 şi la unele AP cum ar fi cele produse de Moeller, sau începând cu unu în cazul AP Trilogi.

Câteva instrucțiuni apar în Tab. 4.1. Dacă după operatorul instrucțiunii apare modificatorul N atunci operandul x este negat. De exemplu, ***LDN x*** încarcă în acumulatorul A valoarea negată a lui x. Câteva aplicații ale instrucțiunilor în scurte secvențe de program tip IL şi LD sunt prezentate în Tab. 4.1

Tab. 4.1 Lista instrucțiunilor principale a unui automat programabil logic şi secvențe tipice de programare

IEC 61131	*Moeller*	*Modificator operand*	*Actiunea asupra operand x. Acesta este negat daca apare modificatorul N*
LD x	***L x***	***N***	***Load x →A***
ST x	***= x***	***N***	***Store A →x***
AND x	***A x***	***N***	***A AND x →A***
OR x	***O x***	***N***	***A OR x →A***
S x	***S x***		***Set x=1 if A=1***
R x	***R x***		***Reset x=0 if A=1***
XOR x	***XO x***		***A XOR x →A***

Nr.	***Program IL***	***Program LD***	***Relația logica***
1	***L I1*** ***A I2*** ***= Q1***	***I1 I2 Q1*** ***|------| |-------| |-----------( )---|***	***H=a.p*** ***Elementul SI***
2	***L I1*** ***O I2*** ***= Q1***	***I1 Q1*** ***|------| |----|----------( )---|*** ***| I2 |*** ***|------| |----|***	***H=a+p*** ***Elementul SAU***
3	***LN I1*** ***=Q1***	***I1 Q1*** ***|------| / |----------( )---|***	***H=/a*** ***Elementul NU cu modificatorul N***
4	***L I1*** ***A I2*** ***=N Q1***	***I1 I2 M1*** ***|------| |-------| |-----------( )---|*** ***| M1 Q1 |*** ***|------| / |----------( )--|***	***H=/(a.p)*** ***Elementul SI-NU***
5	***L I1*** ***O I2*** ***=N Q1***	***I1 M1*** ***|------| |----|----------( )---|*** ***| I2 | |*** ***|------| |----| |*** ***| M1 Q1 |*** ***|------| / |----------( )---|***	***H=/(a+p)*** ***Elementul SAU-NU***
6	***L I1*** ***A I3*** ***= M1*** ***L I2***	***I1 I3 Q1*** ***|------| |----| |----|----------( )---|*** ***| I2 I3 |*** ***|------| |----| / |----|***	***H=a./o+p.o*** ***Folosirea memoriei tampon M***

	AN I3 ***O M1*** ***= Q1***		
8	***L I2*** ***O Q1*** ***A I3*** ***=Q1***	***I3 I2 Q1*** ***|------| |----|------| |---|---( )---|*** ***| | Q1 | |*** ***| |------| |---| |***	***H= /o.(p+k)*** ***Auto-mentinere cu prioritate la oprire.*** ***Oprirea o nu este negata pentru siguranța la avarie[160]***
9	***L I2*** ***S Q1*** ***L I1*** ***R Q1***	***I2 SQ1*** ***|------| |----------( )---|*** ***| I1 RQ1 |*** ***|------| |----------( )---|***	***Bistabil SR cu prioritate la oprire*** ***p = Set*** ***a = Reset***
10	***L I1*** ***R Q1*** ***L I2*** ***S Q1***	***I1 RQ1*** ***|------| |----------( )---|*** ***I2 SQ1*** ***|------| |----------( )---|***	***Bistabil SR cu prioritate la pornire*** ***p = Set*** ***a = Reset***

Tab. 4.2 Configurare APL pentru intrări şi ieşiri

i	***p***	***a***	y_m	y_h
I0.0	I0.1	I0.2	Q0.1	Q0.0
M0.0	M0.1	M0.2	M0.14	M0.15

Tab. 4.3 Configurare APL pentru, stări şi memorii de lucru

x_0^t	x_1^t	x_2^t	$x_0^{t+\Delta}$	$x_1^{t+\Delta}$	$x_2^{t+\Delta}$	***tampon1***	***tampon2***
M0.3	M0.4	M0.5	M0.6	M0.7	M0.8	M0.9	M0.10
Q0.3	Q0.4	Q0.5					

Etapa 7. Elaborarea programului de funcționare a automatului programabil logic sub forma unei liste de instrucțiuni[161]. Pentru exemplul studiat lista de instrucțiuni corespunzătoare automatului programabil logic Klockner Moeller PS3 este prezentată în Tab. 4.4. Se observă că programul are următoarele secțiuni:

- ✓ Achiziția intrărilor (Input scan)
- ✓ Calculul relațiilor logice (Logic scan)
- ✓ Actualizarea variabilelor de stare
- ✓ Calculul variabilelor de ieşire
- ✓ Furnizarea ieşirilor (Output scan)
- ✓ Operațiuni de semnalizare sau testare

[160] „Fail – safety” la intreruperea firului

[161] Instruction List - IL

Tab. 4.4 Programul sub formă de listă de instrucțiuni

Adresa	***Instrucțiunea***	***Comentariu***
000	LI0.0	Achizitia intrărilor
001	=M0.0	
002	LI0.1	
003	=M0.1	
004	LI0.2	
005	M0.2	
006	Lm0.5	Calcul $x_0^{t+\Delta}$
007	ANM0.2	
008	=M0.9	
009	LNM0.1	
010	AM0.3	
011	=M0.10	
012	LM0.9	
013	OM0.10	
014	OM0.0	
015	=M0.6	
016	LM0.3	Calcul $x_1^{t+\Delta}$
017	AM0.11	
018	=M0.9	
019	LNM0.2	
020	AM0.4	
021	=M0.10	
022	LM0.9	
023	OM0.10	
024	ANM0.0	
025	=M0.7	
026	LM0.4	Calcul $x_2^{t+\Delta}$
027	AM0.2	
028	=M0.9	
029	LM0.2	
030	AM0.5	

```
031     =M0.10
032     LM0.9
033     OM0.10
034     ANM0.0
035     =M0.8
036     LM0.6       Actualizare stări
037     =M0.3       x_n^(t+Δ) --> x_n^t
038     LM0.7
039     =M0.4
040     LM0.8
041     =M0.5
042     LM0.4       Calcul ieşiri y_n^t
043     OM0.5
044     =M0.14
045     LM0.3
046     =M0.15
047     LM0.14      Furnizare ieşiri y_n^t
048     =Q0.1
049     LM0.15
050     =Q0.0
051     LM0.3       Semnalizare stări
052     =Q0.3
053     LM0.4
054     =Q0.4
055     LM0.5
056     =Q0.5
```

4.16.3. Metoda schemei desfăşurate

Programul automatului programabil logic poate fi elaborat şi sub forma unei scheme desfăşurate electrice. Metoda este aceiaşi cu cea prezentată în paragraful 4.14.5 dar de data aceasta se folosesc relațiile logice elaborate stabilite prin metoda prezentată în paragraful 4.16.2. Schema desfăşurată obținută pentru APL Klockner Moeller Easy 412 este prezentată în Fig. 4.67 conform standardului DIN şi IEC iar tabelul de configurare corespunzător în Tab. 4.5. Un exemplu de program sub formă de schemă desfăşurată desenată conform standardului american ANSI/CSA se prezintă în paragraful 4.20.2.

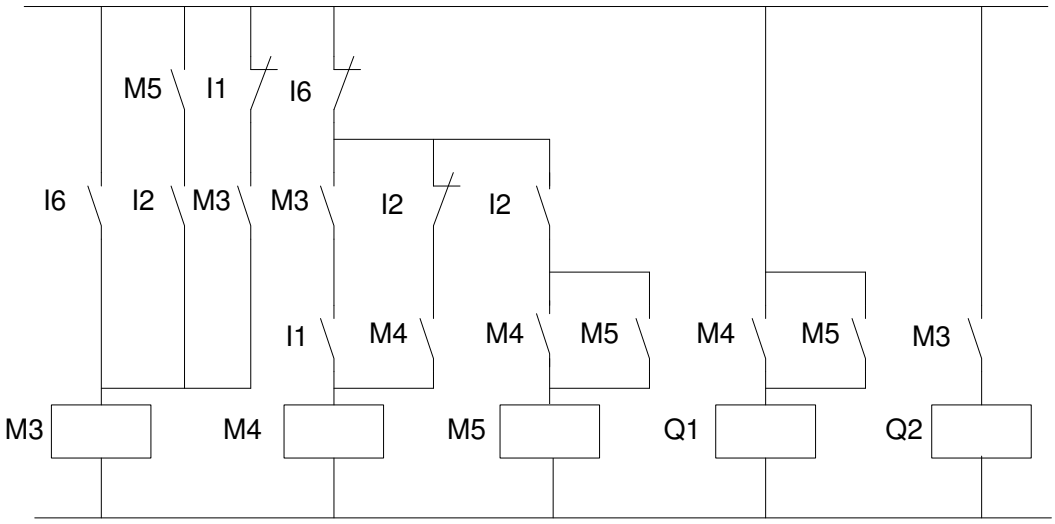


Fig. 4.67 Programul sub formă de schemă desfăşurată pentru automatul Klockner Moeller Easy 412 care opreşte un motor electric după o rotaţie

Tab. 4.5 Tabelul de configurare pentru APL tip Klockner Moeller Easy 412

i	***p***	***a***	$\mathbf{y_m}$	y_h	x_0^t	x_1^t	x_2^t
I6	I1	I2	Q1	Q4	M3	M4	M5

4.16.4. Schema electrica desfăşurată pentru APL

Două automate cu aceiaşi funcţiune logică dar *implementate diferi*[162]*t* sunt reprezentate cu ajutorul unor scheme desfăşurate electrice diferite. În Fig. 4.68 şi Fig. 4.49 se prezintă un exemplu. Se observă că în situaţia implementării cu APL releele nu dispar. Aceasta se datorează faptului că ieşirea APL este formată din contacte de putere mică. Din această cauză se prevede o alimentare a APL şi o alimentare separată în curent continuu a contactelor de intrare şi releelor de ieşire.

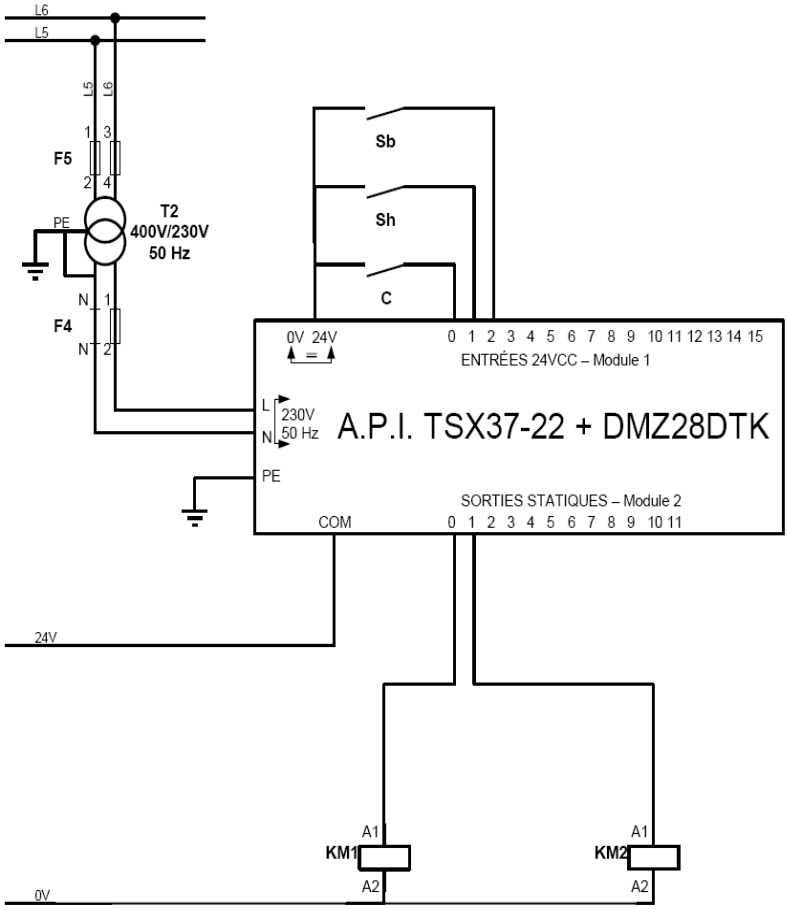


Fig. 4.68 Un exemplu de schemă electrică desfăşurată cu automat programabil logic.

[162] De exemplu, un automat este implementat cu APL iar altul este implementat cu contacte şi relee.

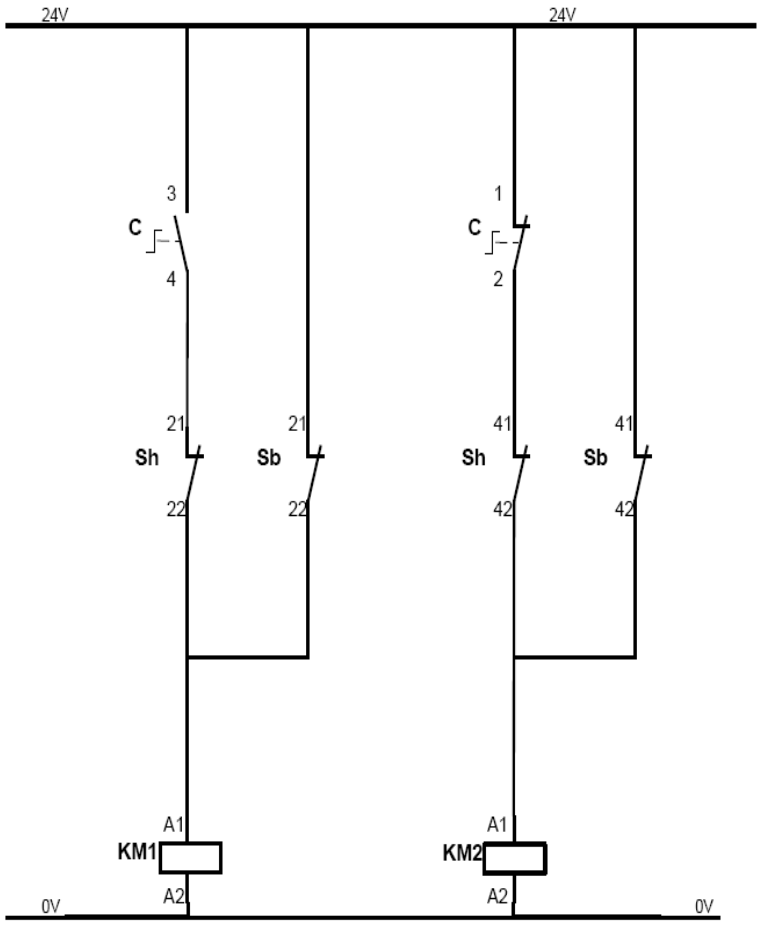


Fig. 4.69 Un exemplu de schemă desfăşurată cu contacte şi relee echivalentă schemei cu APL din Fig. 4.68

4.16.5. Metoda de proiectare pentru APL aplicată la contacte şi relee

Metodele de sinteză a automatelor implementate cu contacte şi relee (ACR) sau cu bistabile care au fost prezentate în paragrafele precedente au dezavantajul că conduc la sisteme discrete cu o funcționare care poate fi înțeleasă mai greu şi prin urmare exploatarea şi

întreținerea lor este mai dificilă. Aceasta se datorează faptului că fiecare stare a automatului este determinată de un număr relativ mare[163] de relee sau bistabile. În plus, metoda de sinteză şi implementare bazată pe metoda diagramelor Karnaugh cu variabile înglobate nu este prea simplă. Este drept că în acest fel rezultă automate cu un număr mic de relee sau bistabile.

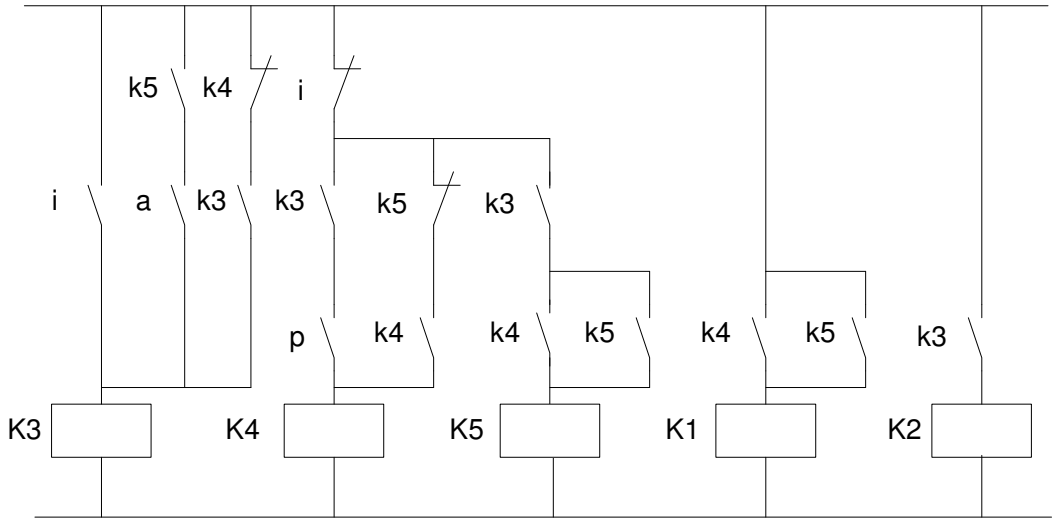


Fig. 4.70 Automatul pentru oprirea unui motor după o rotație implementat cu contacte şi relee după metoda automatului elementar

O alternativă la metodele clasice de proiectare o constitue transformarea schemei desfăşurate care reprezintă programul pentru un APL într-o schemă desfăşurată electrică. Transformarea este imediată şi efectul ei asupra schemei din Fig. 4.67 se observă în Fig. 4.70. Elementele componente ale schemei reprezintă de data aceasta dispozitive fizice: bobine, contacte, butoane, etc. În afară de aceasta mai există o schimbare importantă. Regulile de evoluție a unei rețele Petri, de exemplu cea din Fig. 4.63 b), precizează că la declanşarea unei tranziții toate locațiile precedente se dezactivează iar locațiile posterioare se activează *simultan.* Deoarece condiția de simultaneitate nu poate fi îndeplinită la implementarea cu contacte şi relee este necesar ca mai întâi să se activeze locația posterioară şi apoi să se dezactiveze locația precedentă. Aceasta se poate realiza, *atunci când este cazul*, impunând ca condiția de resetare a unei locații să fie identică cu activarea locației posterioare. Relațiile logice (4.69), (4.70) şi (4.71) se transformă astfel în relațiile logice (4.77), (4.78) şi (4.79) în care numai condițiile de resetare sunt modificate.

[163] În exemplul simplu al automatului pentru oprirea unui motor după o rotație se folosesc două relee sau bistabile pentru caracterizarea fiecărei stări (etape, poziții). La automatele de complexitate medie se folosesc frecvent trei sau patru relee sau bistabile.

$$n=0 \quad s_0 = x_2 \bar{a} \quad r_0 = x_1 \tag{4.77}$$

$$n=1 \quad s_1 = x_0 p \quad r_1 = x_2 \tag{4.78}$$

$$n=2 \quad s_2 = x_1 a \quad r_2 = x_0 \tag{4.79}$$

4.16.6. Proiectarea automatelor pentru rețele Petri simple.

Metodele de proiectare a automatelor implementate cu APL sunt valabile pentru rețele Petri *conforme,* adică echivalente cu un grafcet[164]. Există însă multe rețele Petri conforme cu o structură mai simplă. De exemplu rețeaua Petri conformă tip maşină de stare este caracterizată prin tranziţii în care intră şi ies câte un singur arc. În acest caz declanşarea tranziţiilor pentru condiţiile de resetare nu mai trebuie să fie validată. Acesta este cazul pentru exemplul folosit în paragraful 4.16.2. Cu această observatie conditiile de resetare din (4.69) devine r_0=p. Se observă că înlocuirea acestei condiţii în relaţiile (4.72) şi (4.74) conduc la acelaşi rezultat.

4.16.7. Proiectarea automatelor folosind automate elementare tip RS

Metoda de proiectare a listei de instrucţiuni şi metoda de proiectare a schemei desfăşurate se bazează pe relaţiile (4.72) şi (4.74) care realizează automate elementare cu prioritate la pornire. Automatele programabile logice moderne au în setul lor de instrucţiuni însă şi automatele de tip RS. Dacă se folosesc aceste instrucţiuni sau releele corespunzătoare din metoda schemei desfăşurate sunt suficiente numai condiţiile de setare şi resetare. Prioritatea la pornire este realizată aplicând mai întâi instrucţiunea de resetare R şi apoi instrucţiunea de setare S. Un exemplu este prezentat în paragraful 4.22.5.

Această metodă, contrar aparenţelor, nu conduce la un număr de instrucţiuni sau circuite mai mic în cadrul metodelor de proiectare care folosesc lista de instrucţiuni sau, respectiv, schema desfăşurată. Se pierde însă analogia cu implementarea cu contacte şi relee, foarte intuitivă pentru dezvoltarea şi întreţinerea sistemului. Există însă câteva situaţii specifice în care este recomandabilă, de exemplu atunci când un circuit nu poate să conţină mai mult de trei contacte, cum este cazul pentru APL Easy Relay.

[164] O rețea Petri conformă este echivalentă cu un grafcet. Reciproca nu este valabilă întotdeauna.

4.17. Aspecte constructive ale automatelor cu evenimente discrete

Proiectarea funcțională a automatului se termină odată cu obținerea funcțiilor logice dintre intrările şi ieşirile automatului. Construcția, execuția şi întreținerea automatului se face ajutorul unor scheme, diagrame şi tabele. Obținerea acestora se face în cadrul proiectării constructive.

Schemele, diagramele si tabelele se clasifica după scopul urmărit astfel:

Scheme explicative.

Schemele explicative uşurează studiul si înțelegerea funcționarii unei instalații sau parți de instalație. Se disting:

- scheme funcționale, destinate înțelegerii principiului de funcționare, conținând simboluri si figuri simple. O schema funcțională poate fi o schemă tehnologică, cu automatizare, o schemă bloc, etc.;

- scheme de circulație, destinate înțelegerii în detaliu a funcționării; ele reprezintă prin simboluri o instalație sau o parte din instalație cu conexiunile electrice şi legăturile care intervin în funcționarea sa;

- scheme de echivalență, destinate analizei sau calculului caracteristicilor unui circuit sau element de circuit.

Diagrame sau tabele explicative.

Diagramele sau tabelele explicative sunt destinate uşurării înțelegerii schemelor şi pentru a da informa_ii suplimentare, ca de exemplu:

- diagrama sau tabelul de secvență, care uşurează analiza secțiunilor care se succed într-o ordine determinată;

- diagrama sau tabelul de secvență - timp, care ține cont în plus de valoarea intervalelor de timp între secțiuni succesive.

Scheme de conexiuni sau tabele de conexiuni.

Schemele de conexiuni sunt destinate realizării fizice ăi verificării conexiunilor unei instalații sau echipament. Pot fi:

- scheme de conexiuni interioare;
- scheme de conexiuni exterioare;
- scheme de conectare la borne.

Planuri sau tabele de amplasare.

Un plan sau un tabel de amplasare conține indicații precise despre angrenarea părților unei instalații, de exemplu blocurile terminale, unitățile debroşabile, subansamblele, modulele, etc.

4.17.1. Schema tehnologică cu echipamentul de automatizare

Prima schemă dintr-un proiect de automatizare trebuie să fie schema tehnologică cu echipamentul de automatizare. Fiecare aparat sau echipament al sistemului automat apare pe schema tehnologică sub forma un simboluri convenționale. Un exemplu tipic este prezentat în Fig. 4.71. Se foloses simbolurile din Tab. 4.6.

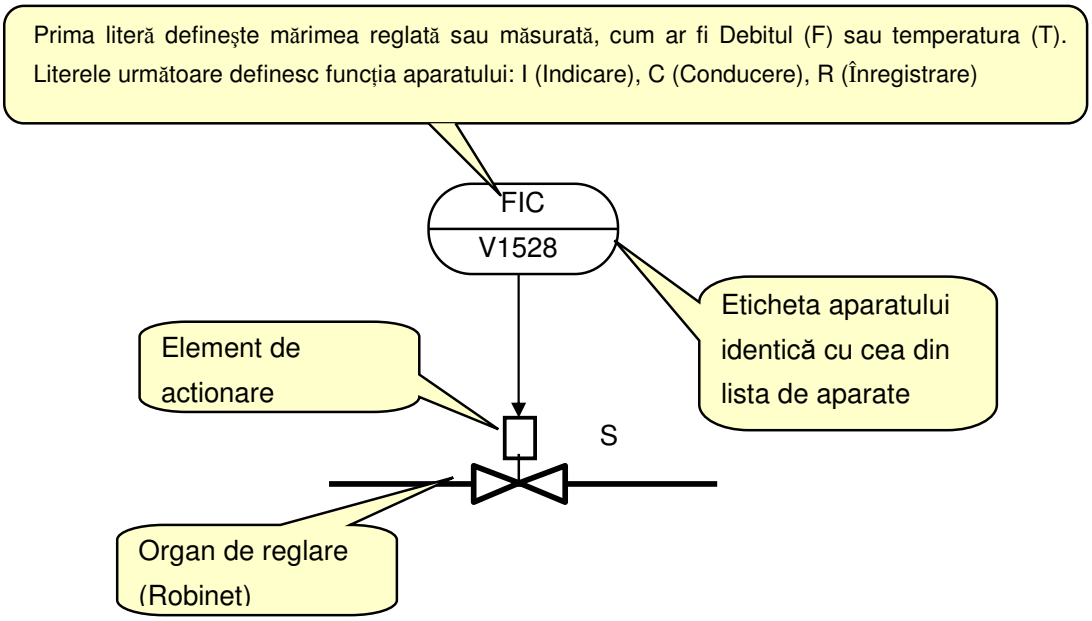

Prima literă defineşte mărimea reglată sau măsurată, cum ar fi Debitul (F) sau temperatura (T). Literele următoare definesc funcţia aparatului: I (Indicare), C (Conducere), R (Înregistrare)
FIC
V1528
Eticheta aparatului identică cu cea din lista de aparate
Element de actionare
S
Organ de reglare (Robinet)

Fig. 4.71 Reprezentarea aparatelor şi echipamentelor de automatizare într-o schemă tehnologică.

Tab. 4.6 Simbolurile aparatelor şi echipamentelor de automatizare

	Montate centralizat	**Montate pe instalaţie**	**Montate într-o locaţie auxiliară**
Aparate	1	2	3
Tablouri	4	5	6
Calculatoare	7	8	9
Automate Programabile	10	11	12

Dimensiunile simbolurilor se pot modifica după necesitate.
ele inaccesibile)în spatele tabloului) se reprezintă cu aceleaşi simboluri dar cu o linie punctată la mijloc. Sursa: Control Engineering şi standardul ISA S5.1.

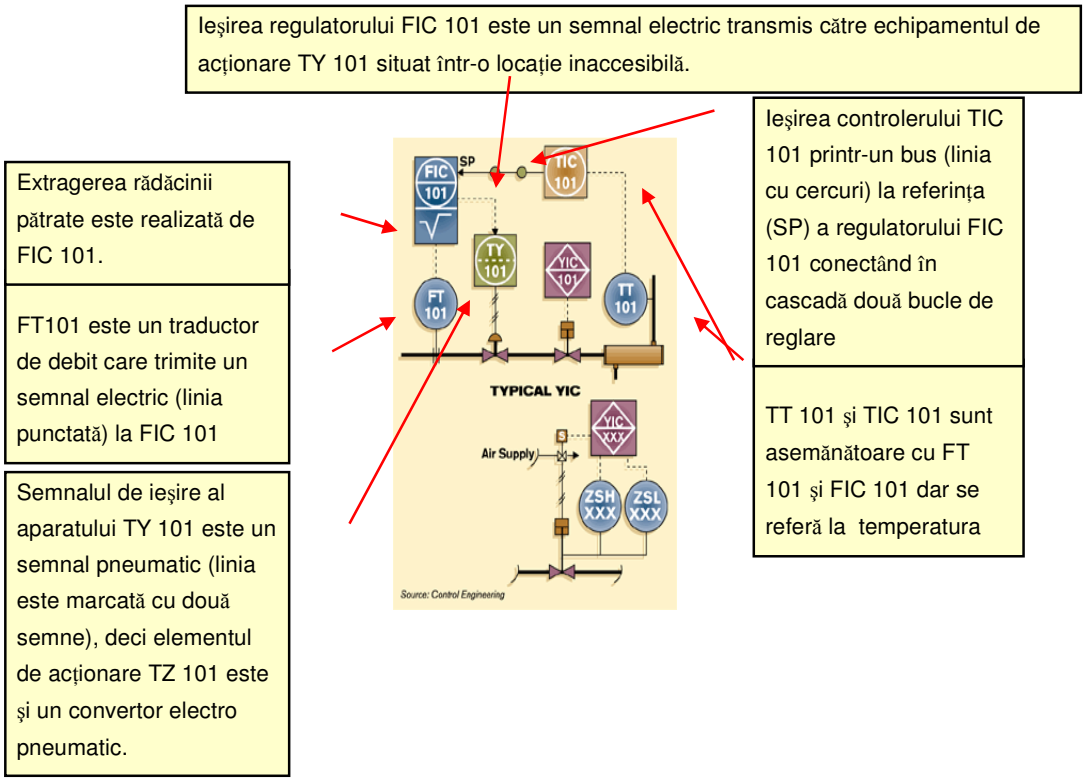

Ieşirea regulatorului FIC 101 este un semnal electric transmis către echipamentul de acționare TY 101 situat într-o locație inaccesibilă.
Extragerea rădăcinii pătrate este realizată de FIC 101.
FT101 este un traductor de debit care trimite un semnal electric (linia punctată) la FIC 101
Semnalul de ieşire al aparatului TY 101 este un semnal pneumatic (linia este marcată cu două semne), deci elementul de acționare TZ 101 este şi un convertor electro pneumatic.
Ieşirea controlerului TIC 101 printr-un bus (linia cu cercuri) la referința (SP) a regulatorului FIC 101 conectând în cascadă două bucle de reglare
TT 101 şi TIC 101 sunt asemănătoare cu FT 101 şi FIC 101 dar se referă la temperatura
SP
FIC 101
TIC 101
TY 101
YIC 101
FT 101
TT 101
TYPICAL YIC
YIC XXX
Air Supply
ZSH XXX
ZSL XXX
Source: Control Engineering

Fig. 4.72 Un exemplu de shemă tehnologică cu echipamentul de automatizare

Tab. 4.7

Simbolul	**Prima literă specifică natura fizică a mărimii prelucrate**	**Următoarele litere indică funcțiile aparatului**
A	Analiză	Alarmă
B	Flacără	Semnificație la alegere
C	Semnificație la alegere	Reglare automată
D	Semnificație la alegere	
E	Tensiune electrică	Element primar
F	Debit	
G	Semnificație la alegere	
H	Acțiune declanşată manual	Maxim

I	Curent electric	Indicale
J	Putere	
K	Timp. Program	Automat
L	Nivel	Minim
M	Umiditate	
N	Semnificație la alegere	Semnificație la alleger
O	Semnificație la alegere	
P	Presiune sau vacuum	
Q	Cantitate	
R	Radioactivitate	Înregistrare
S	Viteză sau frecvență	Comutare
T	Temperatură	Transmitere la distanțî
U	Mai multe variabile	Funcțiuni multiple
V	Vibrație, analiză mecanică	Robinet
W	Greutate sau forță	
X	Neclasificat	
Y	Eveniment, stare	Relee sau elemente de calcul
Z	Poziție, dimensiune	Servomotor sau element final de comandă neclasificat

Două exemple de scheme electrice cu echipamentul de automatizare sunt prezentate în Fig. 4.72 şi Fig. 4.73. Semnificația simbolurilor folosite este explicitată în Tab. 4.7.

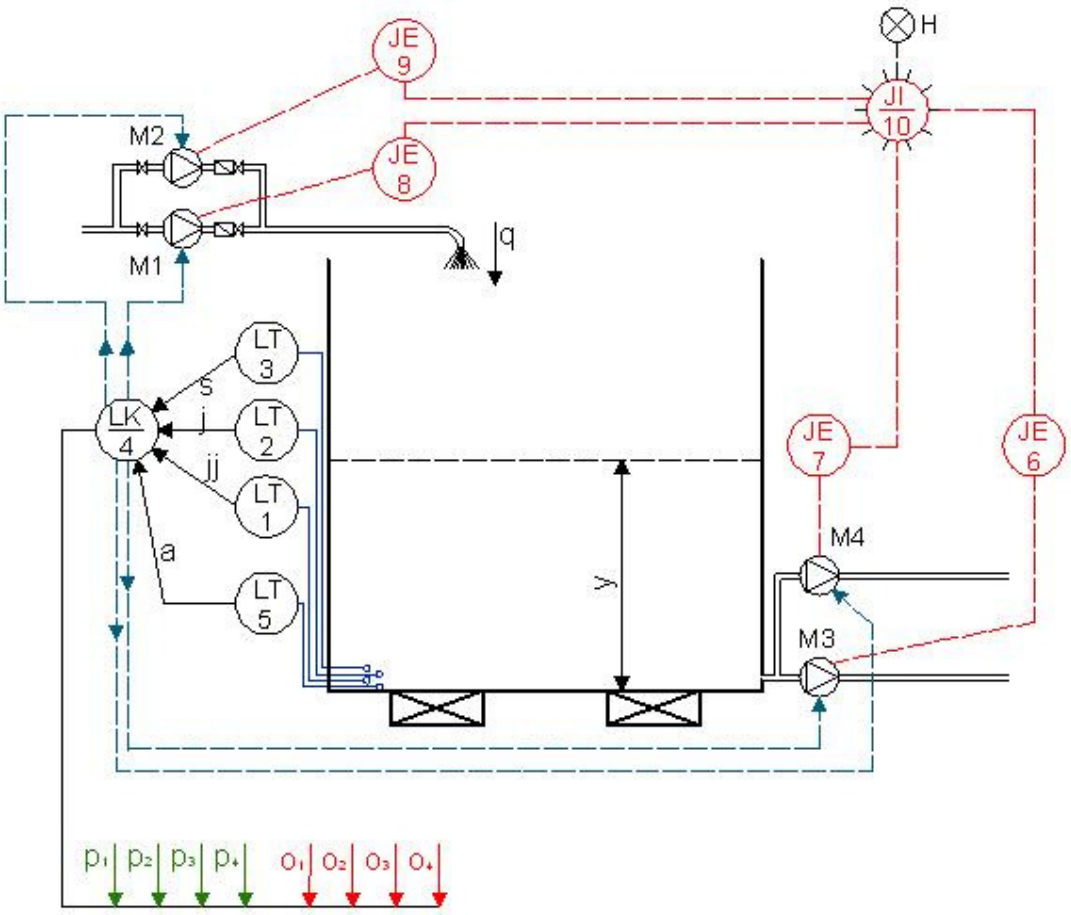


Fig. 4.73 Exemplu de schemă tehnologică cu echipamentul de automatizare

4.17.2. Schema electrică desfăşurată

În reprezentarea schemelor desfăşurate electrice ale automatelor implementate cu contacte şi relee, circuitele electrice sunt aşezate, pentru a asigura înțelegerea uşoară, în ordinea logică a funcționării lor şi sunt desenate de regulă între două linii orizontale reprezentând sursa de alimentare. Pentru reperarea circuitelor, fiecare circuit este marcat, de la stânga la dreapta, cu un număr de ordine înscris la baza schemei. În dreptul circuitelor care conțin bobinele elementelor de comandă sau execuție, se simbolizează fiecare contact acționat şi se indică circuitul în care lucrează contactul respectiv. Se recomandă ca bobinele, lămpile de semnalizare, etc. să fie figurate în apropierea liniei orizontale inferioare. De asemenea, schema electrică de comandă cuprinde, la partea inferioară sau superioară, o manşetă în care se înscrie funcțiunea fiecărui circuit sau grup de circuite.

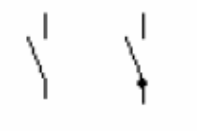
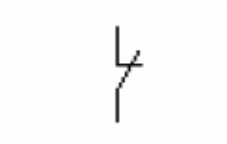
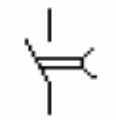
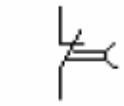
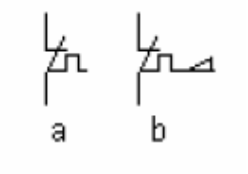
Pentru marcarea aparatelor sau maşinilor se utilizează simboluri literale, Tab. 4.8, iar pentru indicarea bobinelor, butoanelor, contactelor se folosesc simboluri convenționale, Tab. 4.9, intuitive.

Tab. 4.8

Nr. Crt.	**Grupa de aparate sau maşini**	**Simbol**	**Exemple**
1	Condensatoare	C	
2	Dispozitive diverse	E	Dispozitive de iluminat,dispozitive de încălzit, alte dispozitive fără simbol literal propriu
3	Dispozitive de protecţie	F	Siguranţe ,relee de protecţie, declanşatoare mecanice, relee centrifugale ,relee de vânt,dispozitive de descărcare la supratensiune
4	Generatoare (dispozitive de alimentare)	G	Generator, alternator, baterie, dispozitiv de alimentare, convertizor static, convertizor rotativ
5	Dispozitive de semnalizare	H	Avertizoare optice şi acustice, lămpi de semnalizare
6	Relee şi contactoare (altele decât cele de protecţie)	K	
7	Inductanţe	L	Bobine de inducţie, bobine de blocare
8	Motoare	M	
9	Instrumente de măsură , dispozitive de încercare	P	Aparate indicatoare şi înregistratoare, contoare electrice , display , oscilograf , osciloscop , ceas
10	Aparate de comutaţie pentru circuite electrice de forţă	Q	Intreruptor, separator
11	Rezistoare	R	Rezistor cu rezistenţă variabilă , potenţiometru , reostat , şunt , termistor
12	Aparate de comutaţie mecanică pentru circuite electrice	S	Comutator , buton de acţionare, limitator de cursă , termistor
13	Transformatoare	T	
14	Tuburi electronice , semiconductoare	V	Tub electronic , tub cu descărcare în gaze , diodă, tranzistor , tiristor
15	Borne, fişe, socluri	X	Fişe şi prize de conectare , fişe de testare, regletă cu cleme terminale pentru lipit

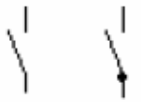
16	Dispozitive mecanice	Y	Frâne, ambreiaje, robineți , electromagneți, maşini electrice de scris, teleimprimatoare

Tab. 4.9

Nr.	Denumirea elementului	Simboluri convenționale
1.	Contact normal deschis	
2.	Contact normal închis	
3.	Contact comutator cu întreruperea circuitului de comutare	
4.	Limitator de cursă a) cu contact normal deschis b) cu contact normal închis	
5.	Contact normal deschis cu temporizare la închidere	
6.	Contact normal deschis cu temporizare la revenire	
7.	Contact normal închis cu temporizare la deschidere	
8.	Contact normal închis cu temporizare la revenire	
9.	Întreruptor ac_ionat prin efect termic a) contact normal închis fără zăvorâre, b) contact normal închis cu zăvorâre	

10.Contact cu releu termic

11. Întreruptor mecanic – semn general

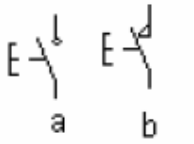


12. Separator

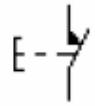
13 Contactor

14 Buton apăsat prin împingere

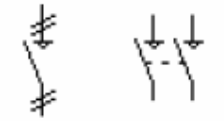
15 Buton acționat prin împingere
a) cu revenire
b) cu reținere



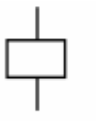
16 Buton cu contact normal închis acționat prin împingere cu revenire automată



17 Întrerupător - separator



18 Bobină releu



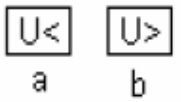
19 Bobină de releu cu temporizare

20 Element de comandă al unui releu termic

21 Releu de tensiune nulă

22 Releu de curent maxim temporizat

23 Releu de tensiune a) minimă, b) maximă

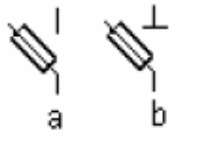


24) Releu de curent a) minim, b) maxim

25 Pilă electrică sau acumulator (linia mai lungă reprezintă polul pozitiv)

26 Siguranță fuzibilă – simbol general

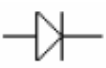
27 Siguranță fuzibilă a) întrerupător, b) separator



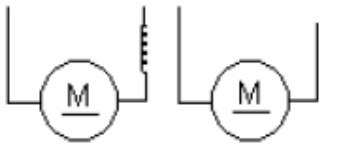
28 Apart indicator (steriscul se înlocuieşte cu un simbol coreespunzător)

29 Lampă de semnalizare

30 Redresor



31 Motor de curent continuu

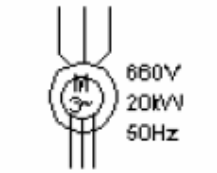


32 Motor asincron cu rotorul în scurtcircuit

33 Motor asincron cu rotorul în scurtcircuit cu şase borne pentru stator

34 Motor asincron trifazat cu rotorul bobinat

35 Motor asincron cu rotorul trifazat şi perii cu indicarea datelor nominate



4.17.3. Tipuri constructive de automate programabile logice

Există trei tipuri constructive de automate programabile logice: micro, mini şi rack (Fig. 4.74)

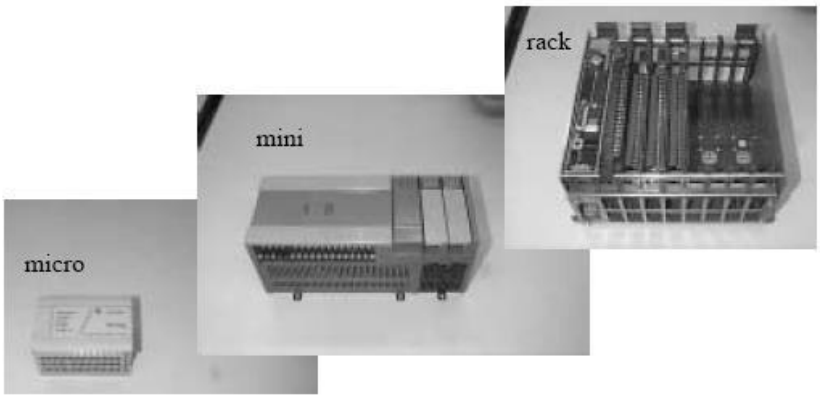


Fig. 4.74 Tipuri constructive de automate programabile logice

Un APL micro este prezentat în Fig. 4.75 iar un APL mini în Fig. 4.76. Automatele programabile logice sunt montate în tablouri, Fig. 4.77, alături se sursele de alimentare, releele şi contactoarele necesare pentru amplificarea de putere a semnalelor şi sistemele de protecție, semnalizare şi interfațare cu operatorul

Fig. 4.75 Platformă experimentală cu APL Moeller Eazy Relay 400

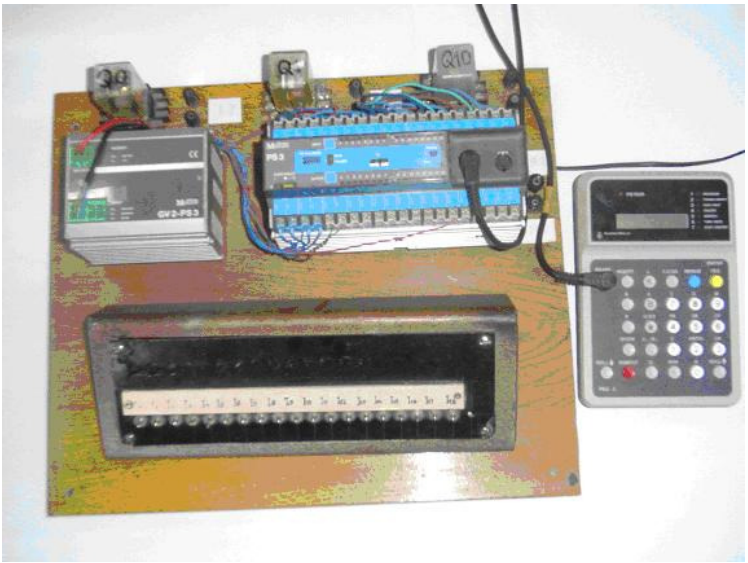


Fig. 4.76 Platformă experimentală cu APL Moeller PS3

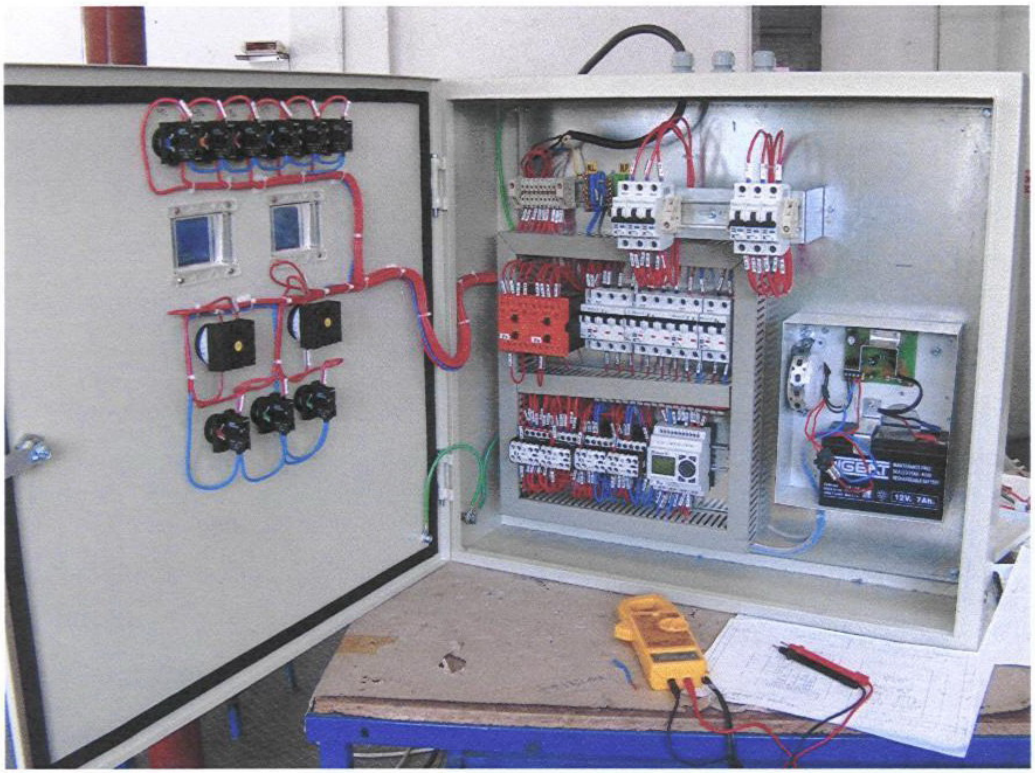


Fig. 4.77 Exemplu de tablou de automatizare

4.18. Automate pentru motoare

Există unele automate care sunt folosite foarte frecvent în practică. Printre acestea se numără automatele pentru pornirea motoarelor electrice asincrone cu rotorul în scurtcircuit. Pentru aceste automate s-au elaborat scheme tip, câteva dintre acestea fiind prezentate în continuare.

4.18.1. Alimentarea automatelor pentru motoare

Automatele pentru motoarele sunt alimentate, în general, monofazat, la 24 V şi în tensiune continuă sau alternativă. Un exemplu este prezentat în *Fig. 4.78*.

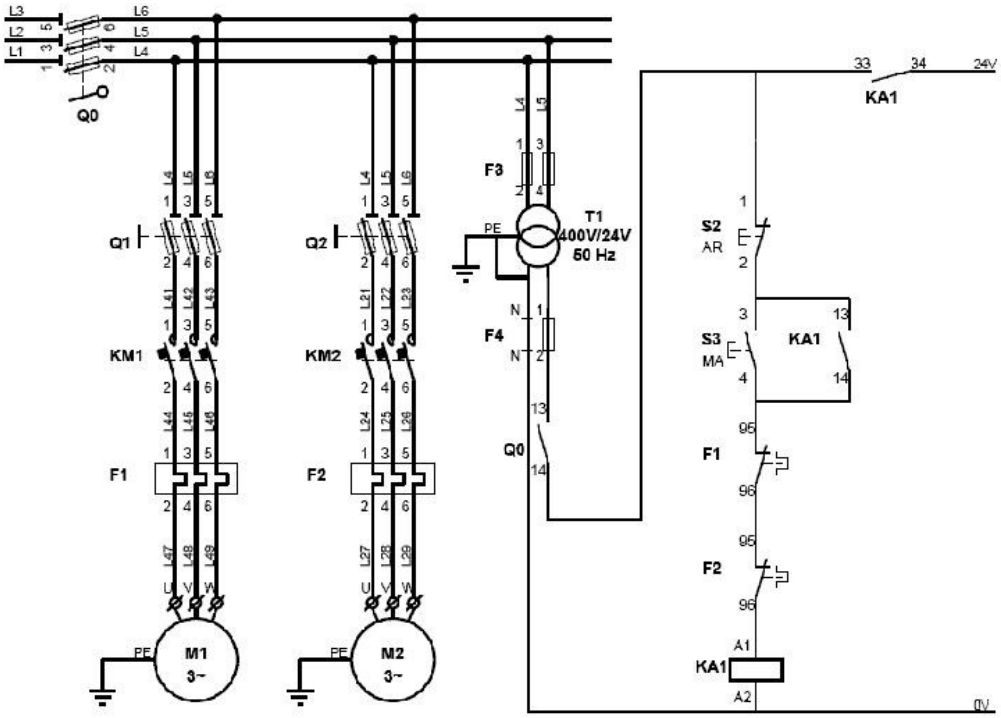


Fig. 4.78 Alimentarea îm curent alternativ şi tensiune continuă a părţii ce comandă automată

4.18.2. Automat pentru pornirea directă

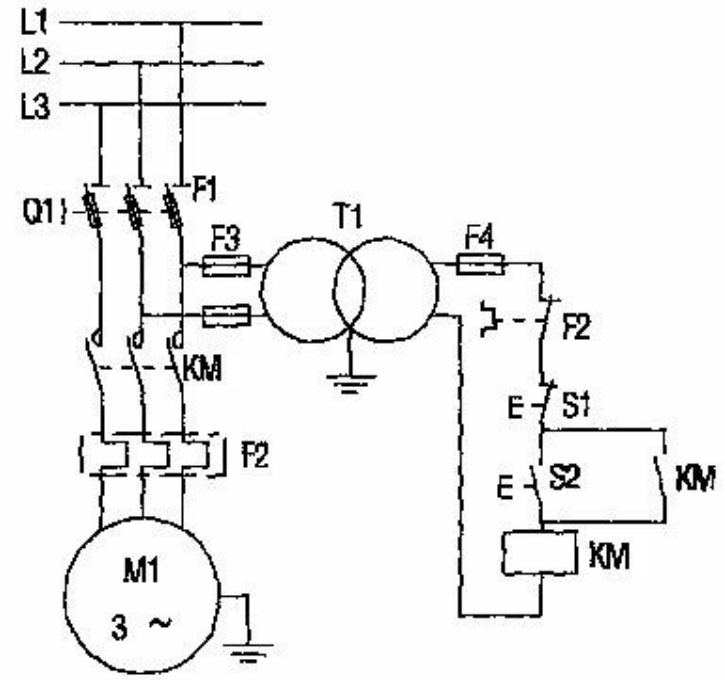


Fig. 4.79 Automat pentru pornirea directă a unui motor

Automatul pentru pornirea directă a motoarelor electrice asincrone cu rotorul în scurtcircuit este un automat elementar cu prioritate la oprire, *Fig. 4.79*.

4.18.3. Automat pentru pornirea stea triunghi

Pentru motoare electrice asincrone cu rotorul în scurtcircuit şi puteri mai mari de 5,5 kW se folosesc automate cu pornirea în stea – triunghi, Fig. 4.80

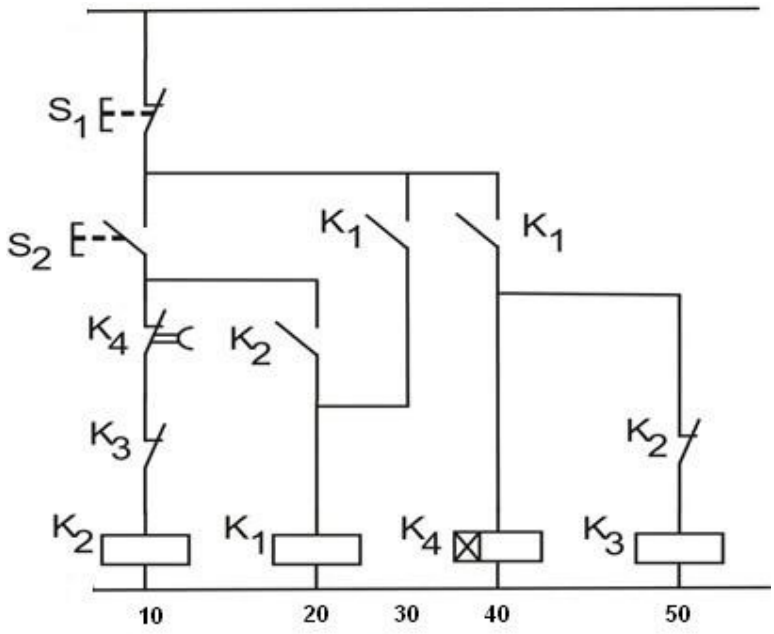


Fig. 4.80 Automat pentru pornirea stea triunghi a unui motor

La apăsarea înm impuls a butonului de pornire S2 se acţionează contactorul K2 de conectare a motorului în stea şi apoi contactorul de conectare a sursei de alimentare trivazate K1. Motorul electric conectat în stea porneşte cu o turaţie redusă. Un contact al lui K1 face automenţinerea conectării sursei de alimentare iar alt contact K1 porneşte releul de timp K4. După câteva secunde se dezacţionează contactorul K2 de conectare a motorului în stea şi se acţionează contactorul K3 pentru conectarea motorului în triunghi. Motorul conectat în triunghi se roteşte cu turaţia nominală. Prin conectarea secvenţială stea – triunghi se evită curenţii mari care apar la pornirea motoarelor electrice asincrone cu rotorul în scurtcircuit.

4.19. Subsisteme specifice ale automatelor

Toate automatele conţin, în general, câteva subsisteme specifice cum ar fi dispozitive de protecţie şi semnalizare automată, dispozitive de achiziţie a intrăprilor şi amplificare a ieşirilor şi dispozitive de comutare manual - automat. Având în vedere importanţa deosebită a acestor elemente componente ale sistemelor automate s-au elaborat pe parcursul timpului subsisteme tipice care sunt introduse, cu unele adaptări, în structura multor sisteme automate.

4.19.1. Sisteme de alimentare

Alimentarea automatelor cu contacte şi relee si a automatelor implementate cu automate programabile logice se face, în general, în curent continuu la o tensiune de 24 V. Contactoarele care sunt folosite drept ampşlificatoare de putere sunt alimentate cu 230 V curent alternativ.

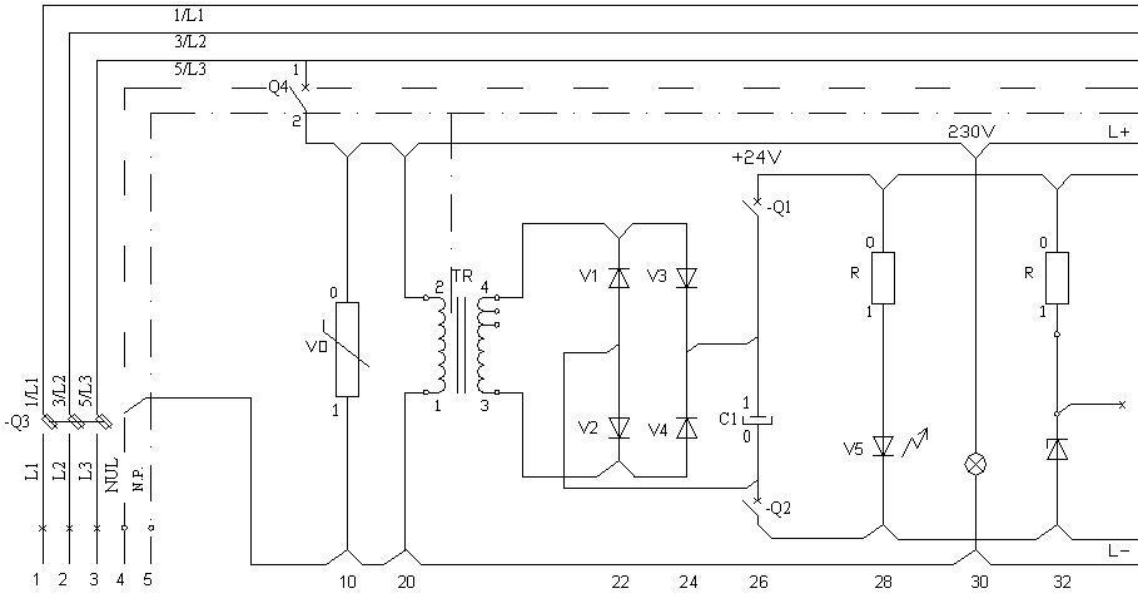


Fig. 4.81 Subsistem tipic de alimentare a automatului în tensiune continuă de 24 V cc şi de alimentare a contactoarelor pentru amplificarea de putere în 230 V ca.

Schema electrică desfăşurată din Fig. 4.81 prezintă op variantă simple de siszem de alimentare a unui automat. Se observă transformatorul, protejat de varistor, redresorul în punte şi stabilizatorul cu diodă Zener. Pentru automatele cu relee nu se folosesc stabilizatoare. De obicei se folosesc stabilizatoare pentru aparate electronice care nu au incluse aceste echipapente, situaţie destul de rară.

4.19.2. Sisteme de achiziție a intrărilor din proces

Intrările pe care automatul mle culege din proces sunt colectate cu ajutorul traductoarelor situate la distanță, pe instalație. Pentru protecția automatului intrarea se face cu ajtorul unor relee intermediare sau a altor dispozitive pentru decuplarea electromagnetică între instalație şi automat. Adeseori se face o filtrare a perturbațiilor de înaltă frecvență cu ajutorul temporizării prodeuse cu ajutorul unor relee de timp.

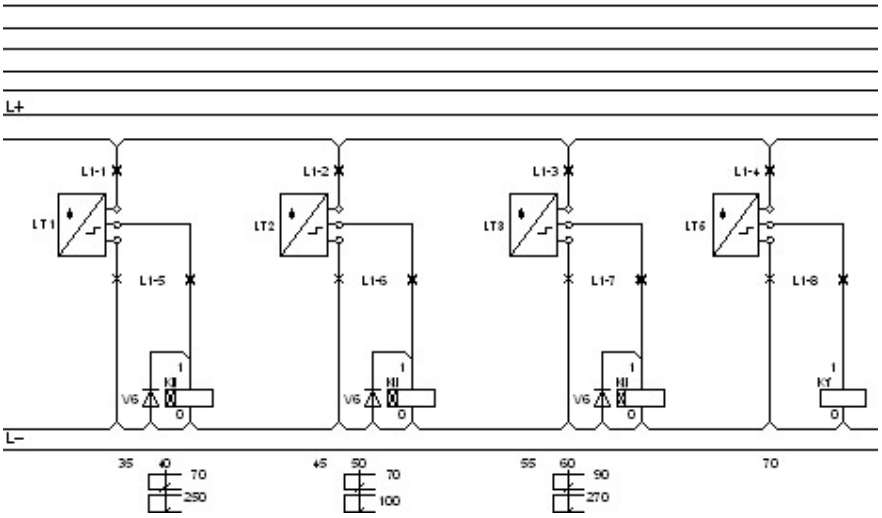


Fig. 4.82 Subsistem de achiziție a intrărilor cu filtrarea perturbațiilo prin temporizare.

4.19.3. Sisteme de conducere automată

Automatele pot fi implementate cu relee sau cu automate programabile logice. În Fig. 4.83 se prezintă o variantă simplă în care apar ambele tipuri de automate.

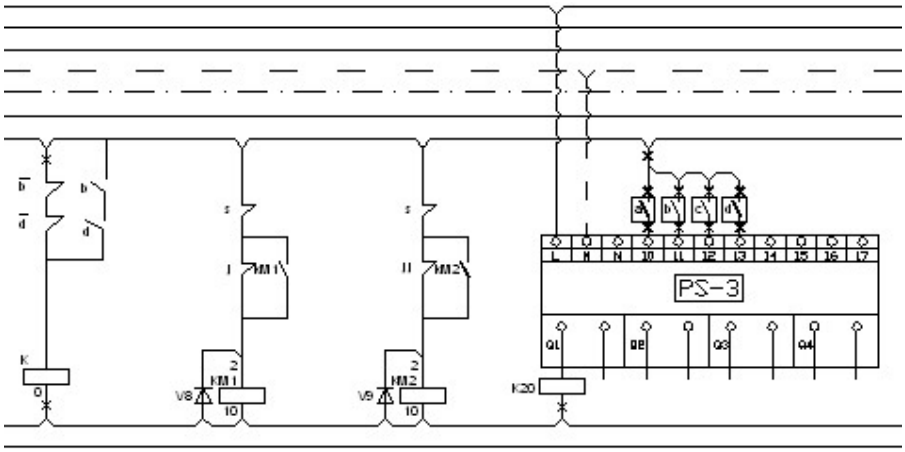


Fig. 4.83 Automate pentru conducerea automată implementate cu relee şi cu APL

4.19.4. Sisteme de conducere manuală

Toate sistemele automate trebuie să aibă şi posibilitatea conducerii manual. O variantă de automat cu comutare pe manual apare în *Fig. 4.84*.

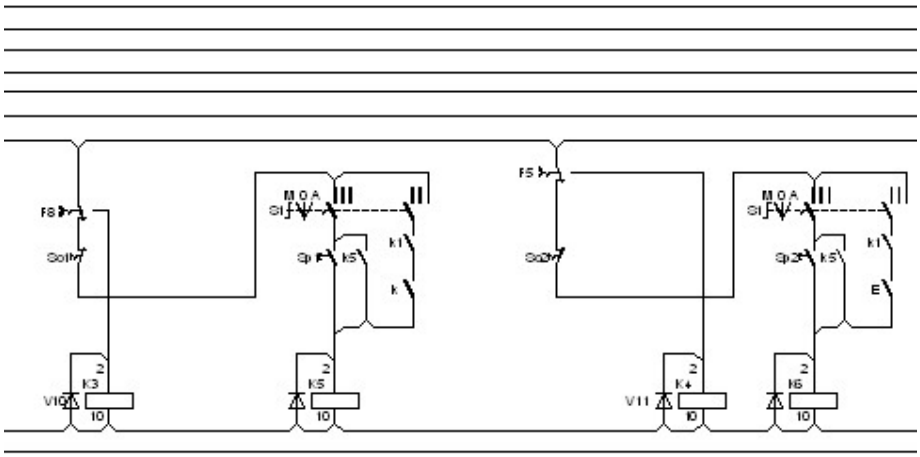


Fig. 4.84 Sistemul de comutare automat - manual

4.19.5. Sisteme de semnalizare automată

Sistemele de semnalizare automat au structuri tipice, asemănătoare cu cea din Fig. 4.85.

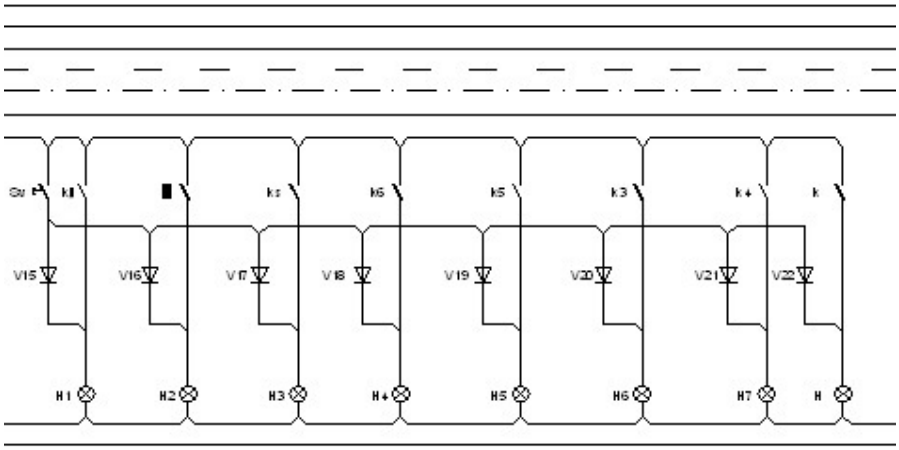


Fig. 4.85 sistemul de semnalizare automată şi de verificare a semnalizării

4.19.6. Sisteme de execuţie şi de protecţie automată

Atât partea de forţă cât şi parte de comandă dintr-un sistem automat trebuie protejate. O schemă desfăşurată simplă esate cea din Fig. 4.86.

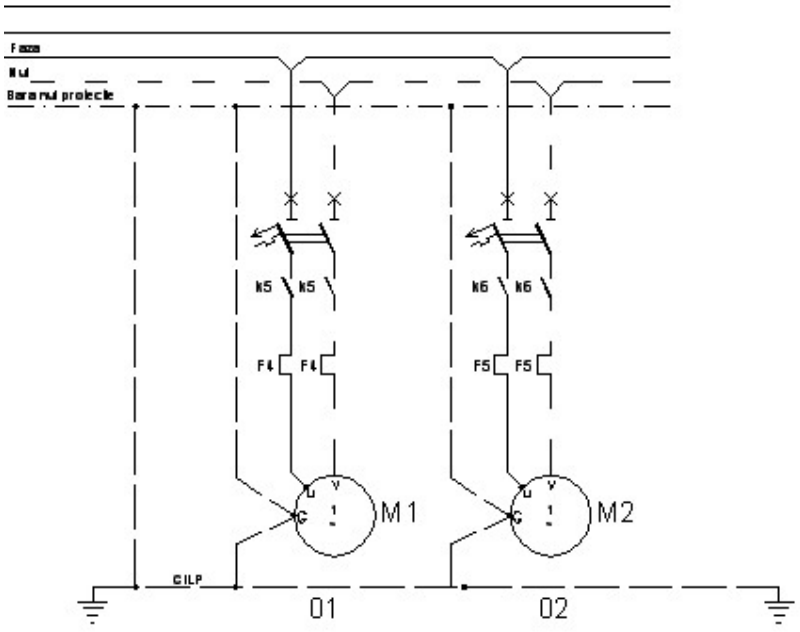


Fig. 4.86 Sistemul de execuţie şi de protecţie automată

4.20. Aplicaţii

Se prezintă în continuare câteva aplicaţii ale automatelor programabile logice.

4.20.1. Automate simple cu APL

În figurile următoare sunt prezentate câteva automate simple împreună cu conexiunile de intrare şi program

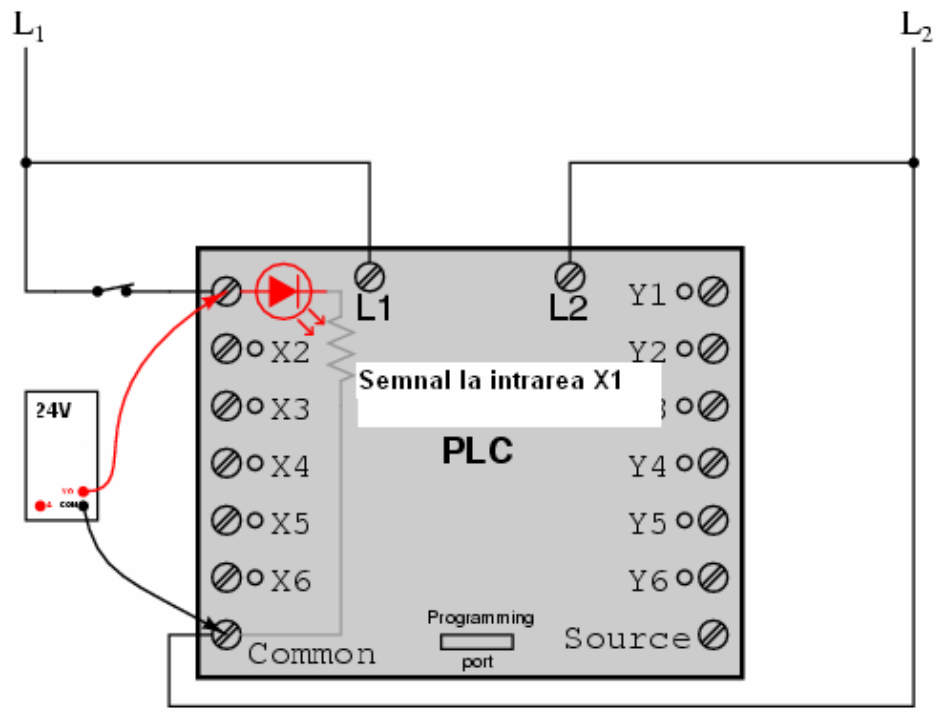


Fig. 4.87 Intrările APL izolate optic

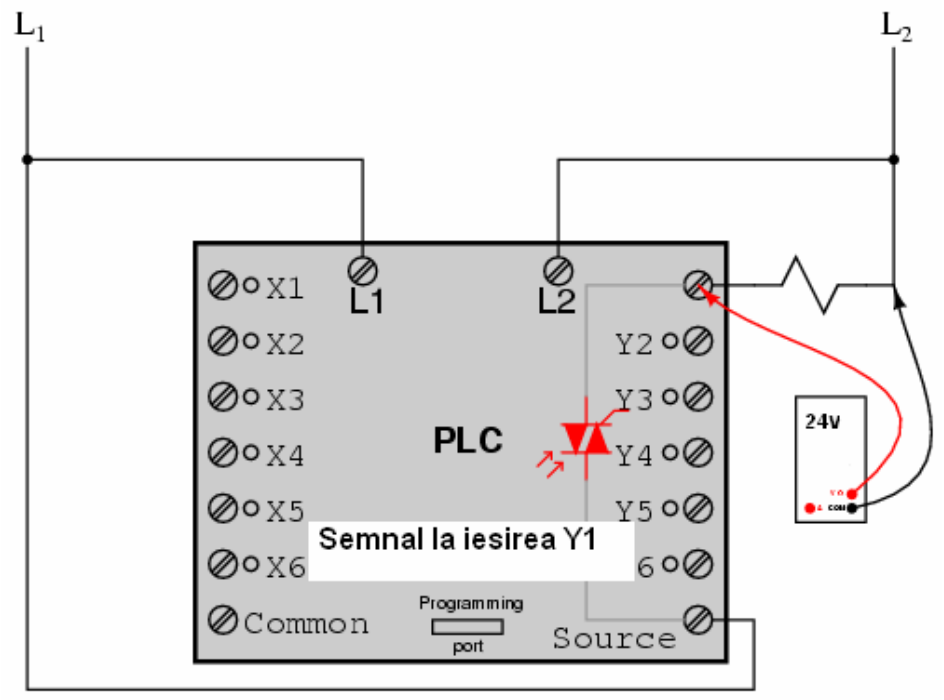


Fig. 4.88 Automatele programabile logice au la ieşire un dispozitiv de comutare.

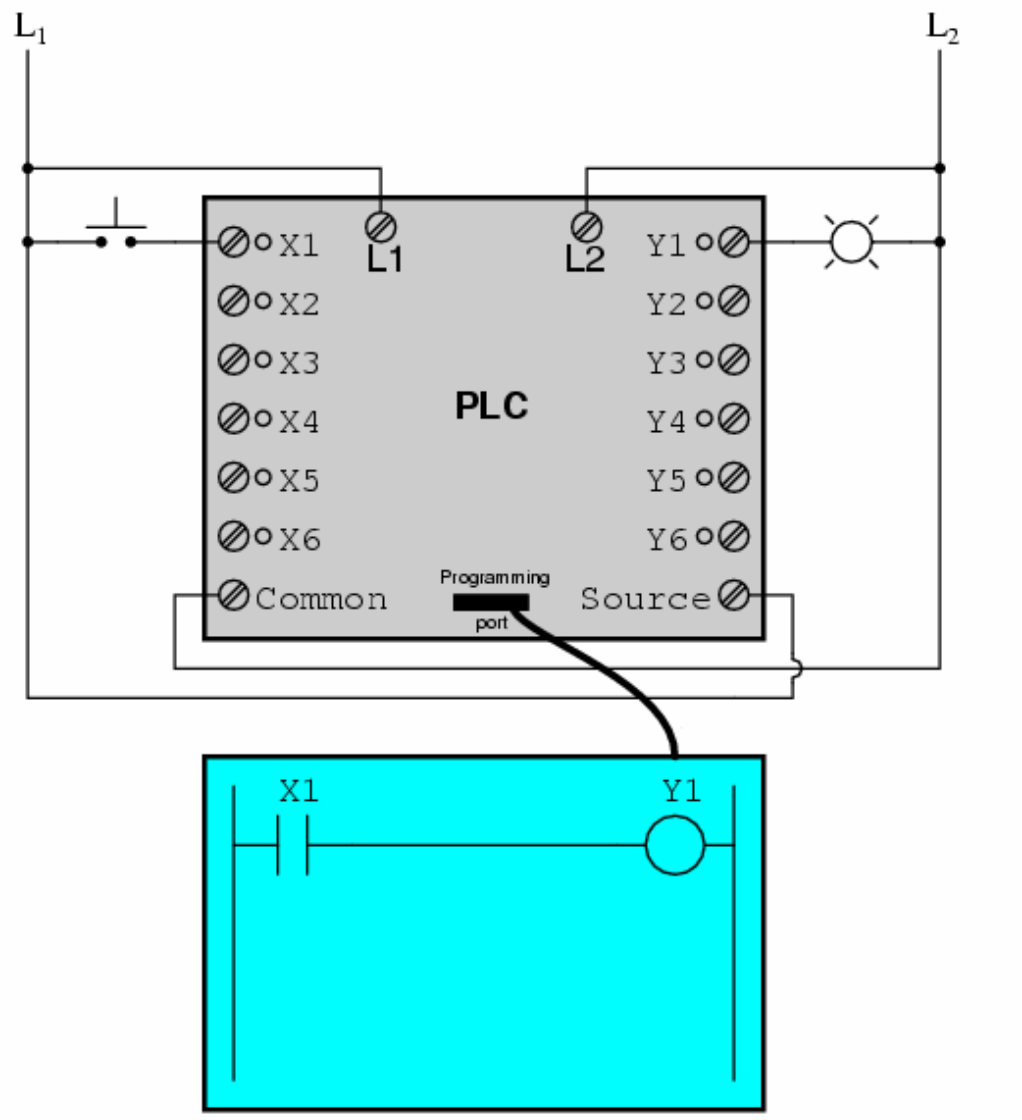


Fig. 4.89 Relația logică dintre intrarea şi ieşirea APL este dată de către program

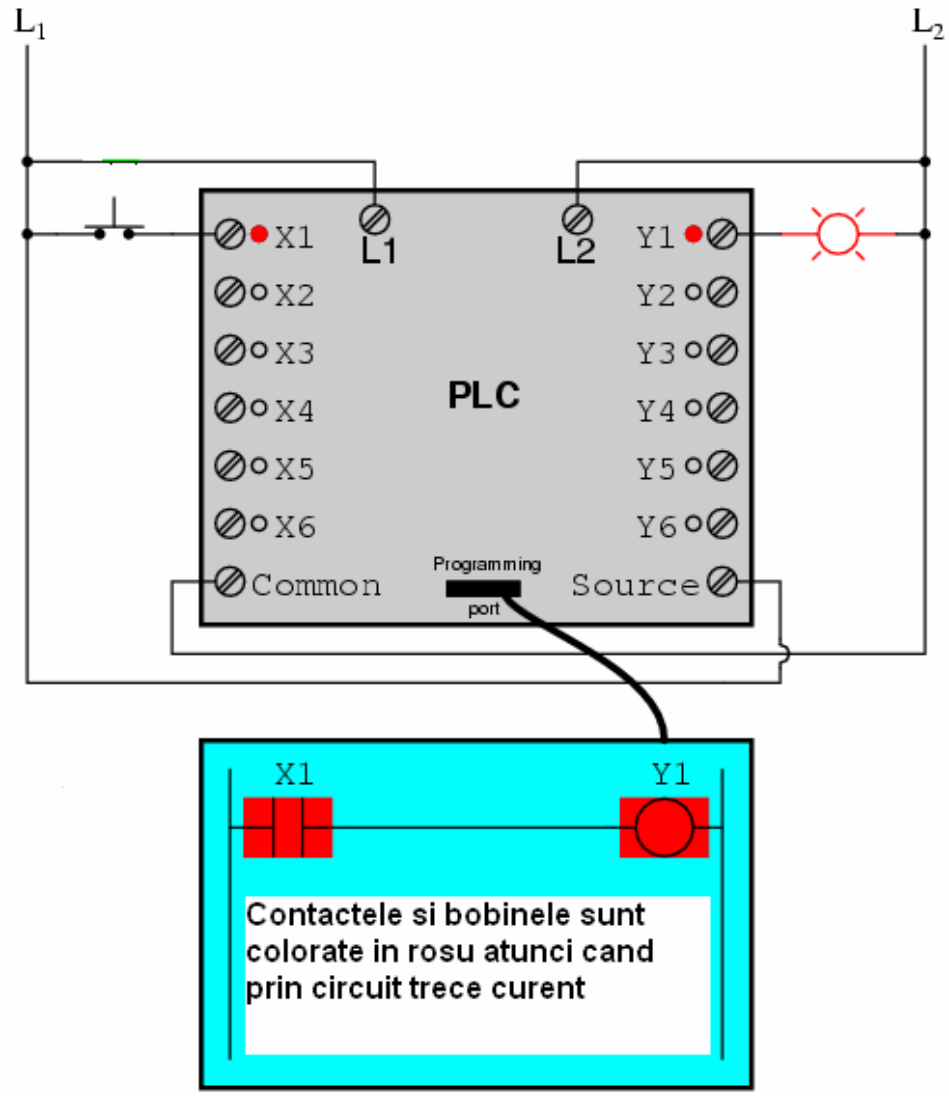


Fig. 4.90 Circuitele din schema desfăşurată prin care trece curent sunt înroşite

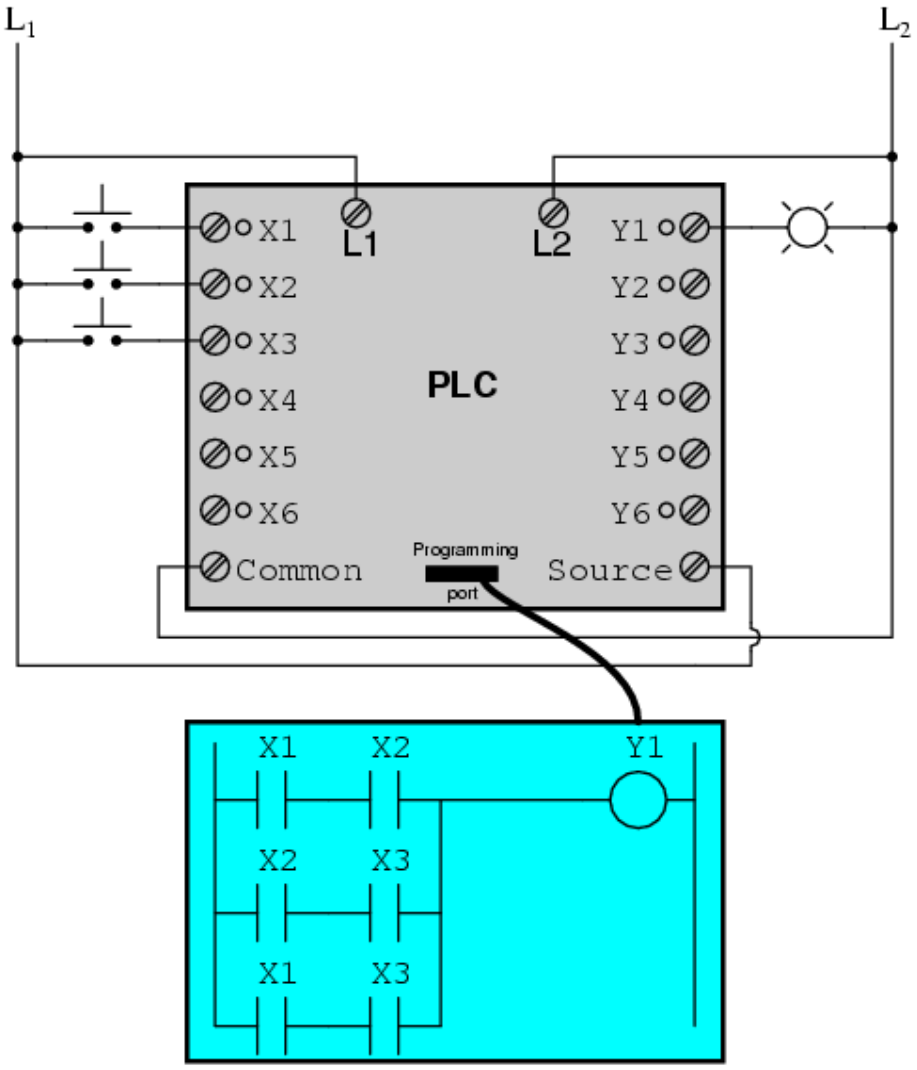


Fig. 4.91 Un SLC cu mai multe intrări şi o ieşire

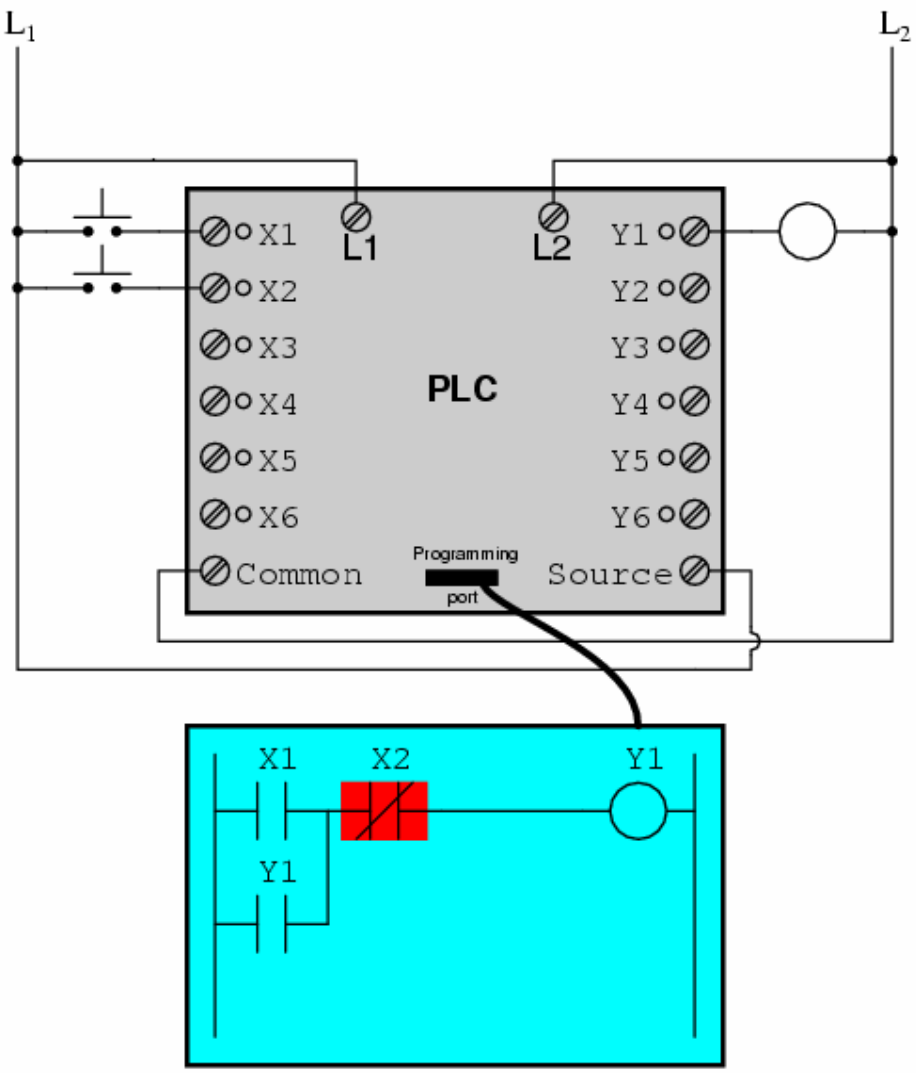


Fig. 4.92 Automatul elementar pentru pornirea si oprirea unui motor

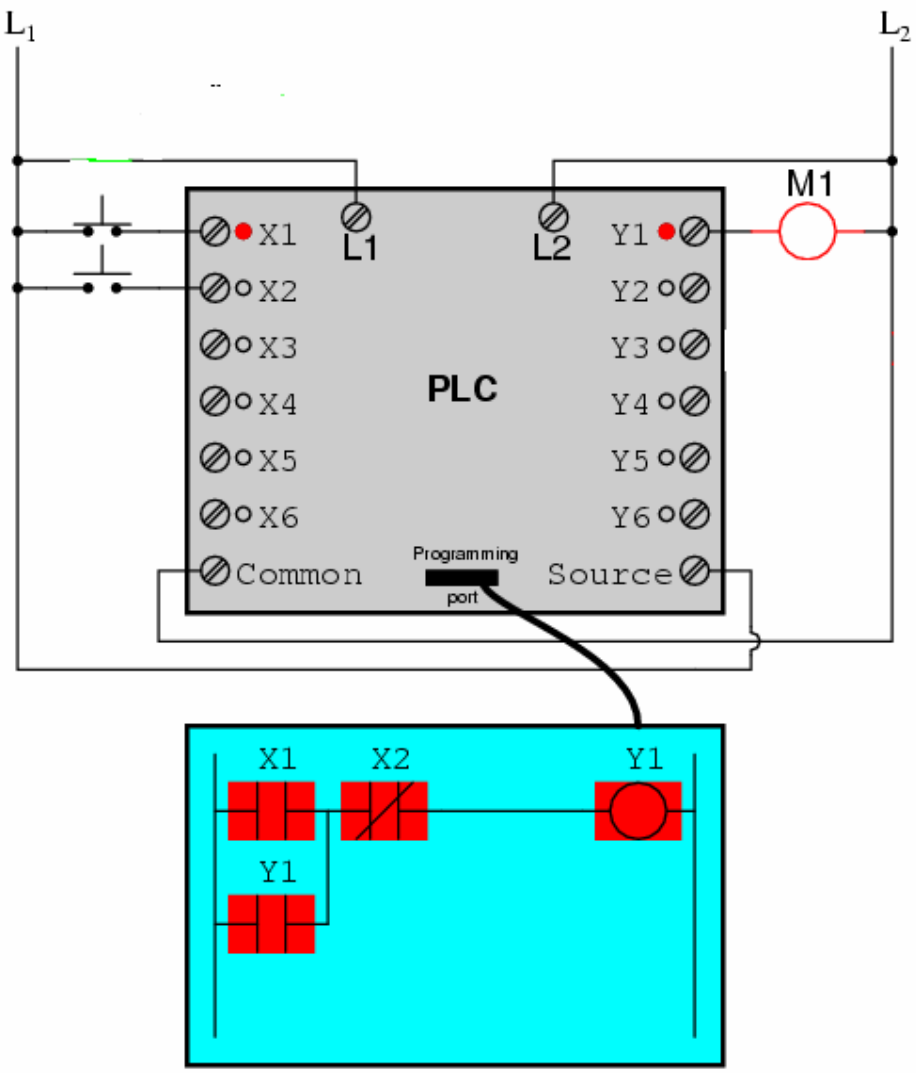


Fig. 4.93 Automatul elementar pentru motor după apăsarea butonului de pornire X1

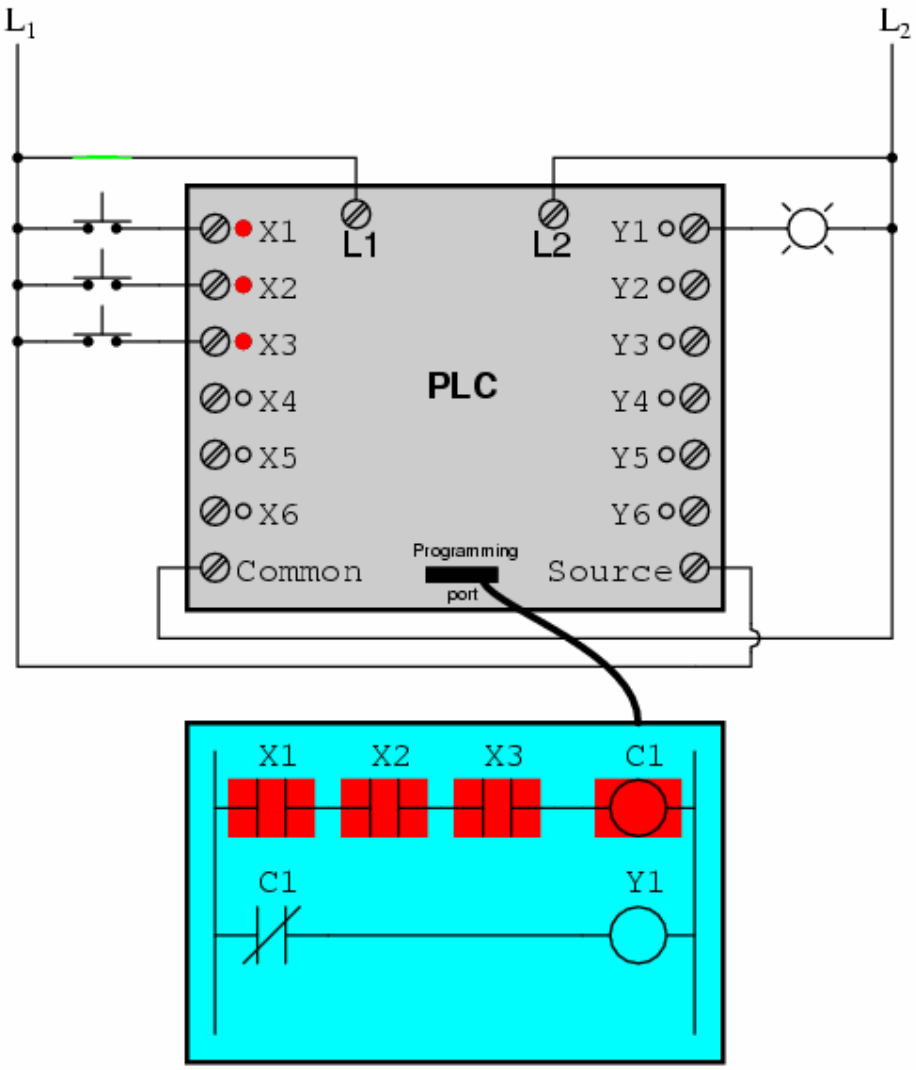


Fig. 4.94 Un sistem de securitate fail-safe

4.20.2. Automat cu APL Tl5 pentru banda rulantă

În continuare se prezintă un exemplu de folosire a automatelor programabile logice Trilogi Tl4 ;o Tl5 pentru implementarea automatului cu evenimente discrete pentru comanda unei benzi rulante dintr-o hală de producție.

Sistemele software Trilogi Tl4 pentru DOS şi Tl5 pentru Windows au mai multe funcțiuni. In continuare vom numii aceste sisteme simulatoare deşi poate nu aceasta este funcțiunea lor cea mai importanta.

- ***Simulatorul DOS Trilogi Tl4***

Pornirea simulatorului sub Windows. Se executa programul simulatorului ***TI4eval exe*** din catalogul Tl4.

Pornirea simulatorului sub DOS. La apariția prompterului DOS se executa programul ***DC*** care este navigatorul numit Dos Commander. Cu ajutorul lui se selectează si se executa programul simulatorului ***TI4eval.exe.***

Simulatorul funcționează in trei moduri de operare diferite:

a) simulare/configurare/compilare,

b) navigare,

c) editare.

In modul de operare ***simulare/configurare/compilare*** este activa bara superioara de comenzi, numita *Meniu* si se realizează următoarele funcțiuni:

1. Simulează in timp real pe calculatoare PC sub sistemul de operare DOS funcționarea anumitor tipuri de automate programabile.
2. Configurează automatul programabil, gestionează comunicarea dintre PC si automatul programabil si compilează programele tip schema desfăşurata întocmite pentru automate programabile in instrucțiuni specifice si le transfera acestora.
3. Compilează schema desfăşurata in instrucțiuni specifice automatului programabil.
4. Gestionează comunicarea dintre Calculatorul PC si automatul programabil

In modul de operare ***navigare*** (Browse) se poate deplasa cu ajutorul tastelor sageti printre circuitele schemei desfăşurate care reprezintă programul automatului programabil simulat. Acum este activa bara de comenzi inferioara.

In modul de lucru ***editare*** se modifica sau se creează circuitele *individuale* ale schemei desfăşurate selectate prin navigare. Din aceasta cauza se poate trece in modul editare numai după ce am intrat in modul navigare si am selectat circuitul dorit. Comenzile se dau tot cu ajutorul barei inferioare care are de data aceasta alt conținut.

La pornirea simulatorului de automat programabil (AP) trebuie sa se încarce (File/*Load)* un program existent sau sa se creeze un program nou *(File/New)!*

Simulatorul are doua bare cu comenzi. Bara superioara este numita *Menu* si conține comenzile simulatorului, configuratorului si compilatorului iar bara inferioara este numita *Browse* si conține comenzile navigatorului sau editorului. Activarea unei bare sau alta se face cu tasta ***ESC***. Aceasta tasta este folosita foarte frecvent si pentru a scăpa de unele comenzi pe

care le solicita programul. Ajutor pentru fiecare comanda din cele doua bare se obține cu tasta *F1*.

Pentru navigarea printre comenzile simulatorului se recomanda folosirea tastelor cu săgeți, ESC, TAB si nu a mauselui.

Oprire. Se foloseşte comanda *File/Quit* sau se apasă simultan tastele *Alt-X*. Pentru aceasta este necesar ca bara superioara de comenzi (Menu) sa fie activa.

- ***Încărcarea unui program existent.***

Se foloseşte comanda *File/Load* din bara superioara de comenzi. Simulatorul solicita numele programului, Daca se foloseşte astericul (wildcard) in numele programului, de exemplu *.pc4, se deschide o fereastra si se poate naviga in directorul curent pentru alegerea programului dorit. Un exemplu este: *autpor.pc4* sau *demo.pc4*.

- ***Elaborarea unei scheme desfăşurate noi.***

Se foloseşte comanda *File/New*. cu care se trece in modul de lucru navigare in care este activa bara inferioara de comenzi. Doua comenzi sunt mai importante. Cu Tasta Ins se activează meniul *Circuit* iar cu tasta *spbar* se activează modul de lucru *editare* care are o bara inferioara proprie de comenzi. Aceste comenzi pot adaugă circuite noi la schema desfăşurata. Cu tasta ESC se poate reveni in celelalte moduri de lucru: simulare/configurare sau navigare.

- ***Simularea funcționarii unui automat programabil.***

Se încarcă sau se elaborează schema desfăşurata care constitue programul automatului programabil. Se trece, daca este necesar, ca ajutorul tastei ESC, in regimul simulare/configurare/compilare in care este activa bara superioara Menu de comenzi. Se executa comanda Simulate/Continue run sau se apasă tasta F9. Apare un panou in care sunt vizibile toate contactele si bobinele automatului programabil. Cu ajutorul săgeților se pot selecta intrările. Valoarea fiecărei intrări (contact) poate fi basculata (trecuta din starea "acționat" in starea "neacționat" şi invers) cu comanda in nivel[165] ***Enter*** sau comanda in impuls[166] ***Ctrl-Enter***. Starea intrărilor si bobinelor este indicata de prezenta unui LED. Daca in aceasta situație se trece cu ajutorul tastei ESC in modul de lucru ***navigare*** se observa starea contactelor si bobinelor, acționate sau neacționate.

- ***Configurarea automatului programabil.***

Înainte de elaborarea unui nou program sub forma de schema desfăşurata trebuie sa se configureze automatul programabil, adică sa se dea denumiri intrărilor si bobinelor. Aceasta se face in modul de lucru simulare/configurare/compilare cu comenzile Edit/Input, Edit/Output, Edit/Relay, etc.

- ***Introducerea unui comentariu.***

Se trece in modul navigare (cu tasta Esc) in care este activa bara inferioara de comenzi si se executa comanda Ins/Put coments.

---

[165] Comanda se memorează.

[166] Comanda nu se memorează.

- ***Editarea unei scheme desfăşurate.***

Pentru a folosi acest mod de lucru este necesar sa existe o schema desfăşurata obţinuta prin comenzile File/Load sau File/New ale barei superioare. Când se creează un program nou schema este goala, nu are nici un circuit. Se selectează circuitul care urmează sa fie editat. Daca schema este goala acesta este in mod implicit circuitul #1. Se trece in modul de operare editare cu comanda Spbar. Cursorul începe sa pâlpâie. Se deplasează cursorul cu ajutorul tastelor săgeţi selectând elementele existente de circuit daca acestea exista. Odată elementul de circuit selectat se selectează cu ajutorul tastei Tab poziţia in care se doreşte introducerea unui element de circuit, contact sau bobina, de exemplu. Se apăsa tasta Ins. Apare un meniu din care se poate selecta tipul elementului care sa fie introdus. Semnificaţia fiecărui simbol se obţine solicitând ajutor cu tasta F1. Pe scurt sunt elemente de circuit (contacte normal deschise, contacte normal închise sau bobine) care pot fi conectate in serie, in paralel sau in paralel/serie. Conectarea serie sau paralel nu pune probleme. Eventual câteva încercări ne familiarizează cu procedura. Mai dificila este conectarea paralel serie. Câteva îndrumări pentru realizarea unei astfel de conexiuni. Se determina punctul din care trebuie sa plece ramura paralela. Pentru aceasta se selectează mai întâi cu ajutorul tastelor săgeţi un element si apoi cu tasta Tab punctul din stânga sau dreapta a acestui element. Daca exista mai multe ramuri paralele in circuitul editat se selectează cu tastele săgeţi si tasta Tab punctul de *conexiune* a ramurilor dorit care este diferit *borna* unui element. Se apăsa tasta Ins pentru apariţia meniului Ins Element. Din acest meniu se selectează elementul cu numărul 5,6 sau 10. Cursorul se transforma intr-un x indicând punctul de pornire al ramurei paralele. Cu tastele săgeţi se deplasează cursorul in poziţia in care se doreşte sa fie punctul terminal al ramurei paralele. Din tabela de configurare dorita (de obicei tabela Input) se alege denumirea elementului (contactului). Daca nu apare tabela dorita (de exemplu Input) cu tastele săgeţi se navighează pana când apare aceasta. Se apasă Enter si elementul ales este conectat in paralel intre punctele dorite. Daca dorim ca in *serie* cu acest element sa introducem un altul se selectează elementul, apoi unul din capetele acestuia si in final se apasă tasta Ins care declanşează secvenţa de operaţii necesara.

- ***Prezentarea tabelului de configurare si a schemei desfăşurate.***

In modul de operare simulare/configurare/compilare se executa comenzile Print/I/O Tables si Print/Label-name Logic. Se poate alege tipărirea intr-un fişier sau la imprimanta. Fişierele rezultante sunt de tip text.

- ***Un exemplu de program sub forma de schema desfăşurata electrica.***

In continuare se prezintă tabela de configurare si programul AP sub forma de schema desfăşurata pentru automatul cu evenimente discrete care conduce o banda rulanta dintr-o hala de producţie. Pentru simulare se observa ca toate intrările sunt de tip impuls si acţionarea lor se face prin apăsarea tastelor Ctrl – Enter. La apăsarea in impuls a doua dintre butoanele a, b sau c trebuie sa se aprindă lampa de semnalizare L avertizând pe al treilea muncitor ca operaţiile celorlalţi doi muncitori sunt terminate si trebuie sa se grăbească. La apăsarea in impuls, de către muncitori, a tuturor celor trei butoane trebuie sa pornească motorul M care

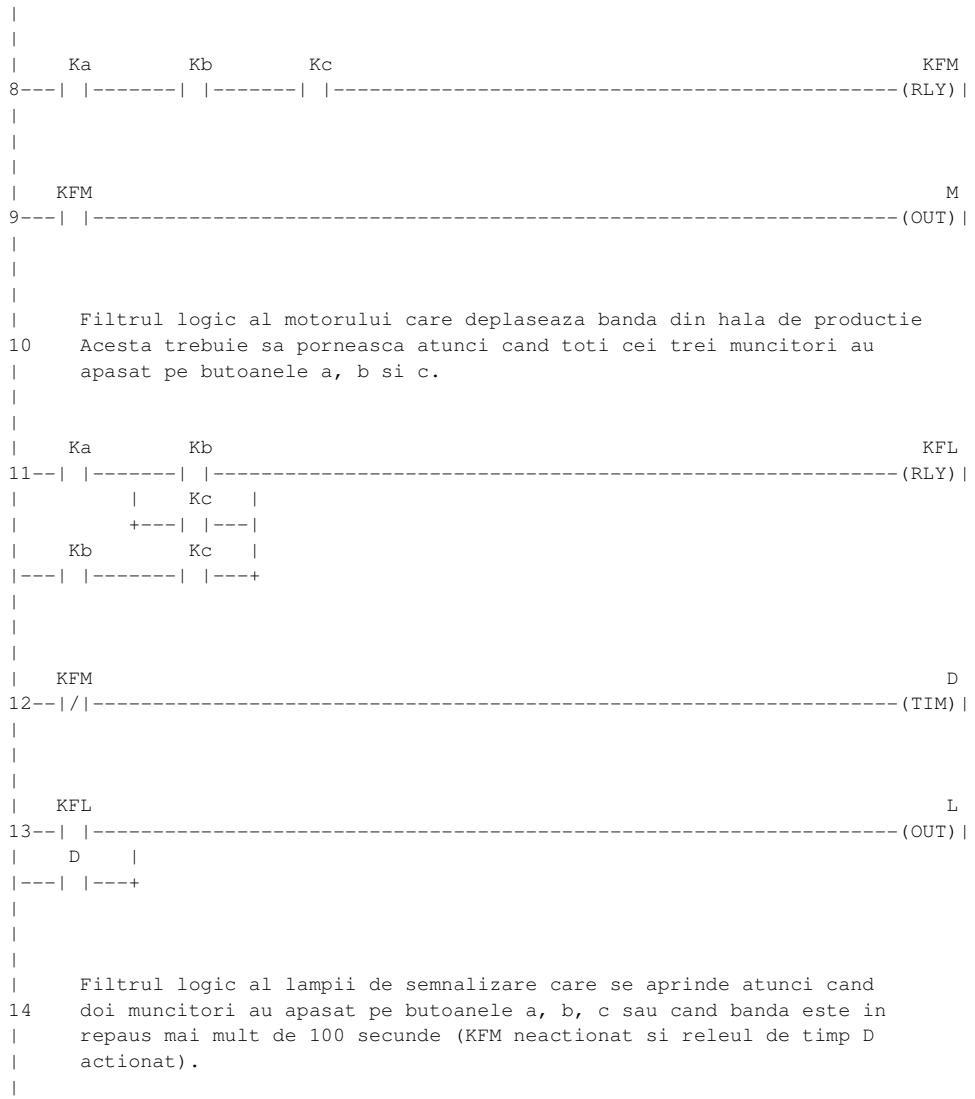
acționează banda rulanta. Când banda a avansat intr-o noua poziție un palpator acționează in impuls contactul traductorului t. Sistemul revine in situația inițiala cu lampa stinsa si motorul oprit. Atunci când banda rulanta este in repaus mai mult de 100 secunde (motorul nu funcționează) se aprinde lampa L care semnalizează ca nu se lucrează in ritmul normal. Daca se apasă tasta Esc in timpul simulării se poate observa pe schema desfăşurata starea contactelor si a bobinelor.

Combined I/O Definition Table

I/O#	Input	Output	Relay	Timer	Value	Counter	Value
1	a	L		D	1000		
2	b	M					
3	c		Ka				
4	t		Kb				
5			Kc				
6			KFL				
7			KFM				
8							

BANDA.PC4

```
|     Automatul cu evenimente discrete pentru comanda unei benzi rulante
1     dintr-o hala de productie.
|     ************************************************************
|
|
|     Trei muncitori apasa butoanele in impuls a, b si c  cand au terminat
2     operatia la banda. Daca au apasat butoanele doi dintre ei automatul
|     aprinde lampa de semnalizare L. Daca au apasat trei automatul porneste
|     motorul M care avanseaza banda cu o pozitie.
|
|     Contactul t al traductorului de pozitie se actioneaza cand banda
3     a ajuns in noua pozitie. Automatul opreste lampa L si motorul M si
|     asteapta noile impulsuri a, b si c de la muncitori.
|
|
|    t         a                                                        Ka
4---|/|-------| |----------------------------------------------------------(RLY)|
|         |    Ka   |
|         +---| |---+
|
|
|
|    t         b                                                        Kb
5---|/|-------| |----------------------------------------------------------(RLY)|
|         |    Kb   |
|         +---| |---+
|
|
|
|    t         c                                                        Kc
6---|/|-------| |----------------------------------------------------------(RLY)|
|         |    Kc   |
|         +---| |---+
|
|
|
|     Se memoreaza impulsurile a, b si c trimise automatului de catre
7     muncitorii la banda rulanta.
|
```



- ***Simulatorul Windows Tl5***

Modul de operare a acestui simulator pe calculatorul PC local este intuitiv şi asemănător cu varianta DOS Tl4. Varianta educativă a ecestui simulator nu permite folosirea simulatorului sub forma unui server pentru controlarea la distanţăa unui APL, pe intranet şi internet.

La fel ca şi la varianta Dos apare o dificultate la elaborarea programului sub formă de schemă desfăşurată (LD) atunci când se introduce o ramură paralelă care conţine mai multe contacte serie. În această variantă se folosesc contacte de tipul 5 la care trebuie să se stabilească distinct cele două puncte de conectare. Pentru aceasta se observă că atunci când se selectează un punct cu ajutorul mauseului sau a săgeţilor apare borna contactului sub forma unui pătrat sau punctul de conexiune sub forma unei pete. După selectarea unui contact cu mauseul transformarea bornei în punct de conexiune cu ajutorul tastelor săgeţi se face conectarea unui contact de tip 5 iar punctul de conexiune opus se transformă într-un x. Punctele de conexiune se găsesc pe ramura principală a circuitului pe care se găseşte bobina.

După apariția primei conexiuni sub forma unui x se determină pe circuitul principal cu ajutorul tastelor săgeți al doilea punct de conexiune , care apare sub forma unei pete şi se conectează din nou componenta 5.

4.20.3. Automat cu APL pentru motorul reversibil cu inițializare

Calculul automatului servomotorului reversibil cu inițializare al robinetului de reglare

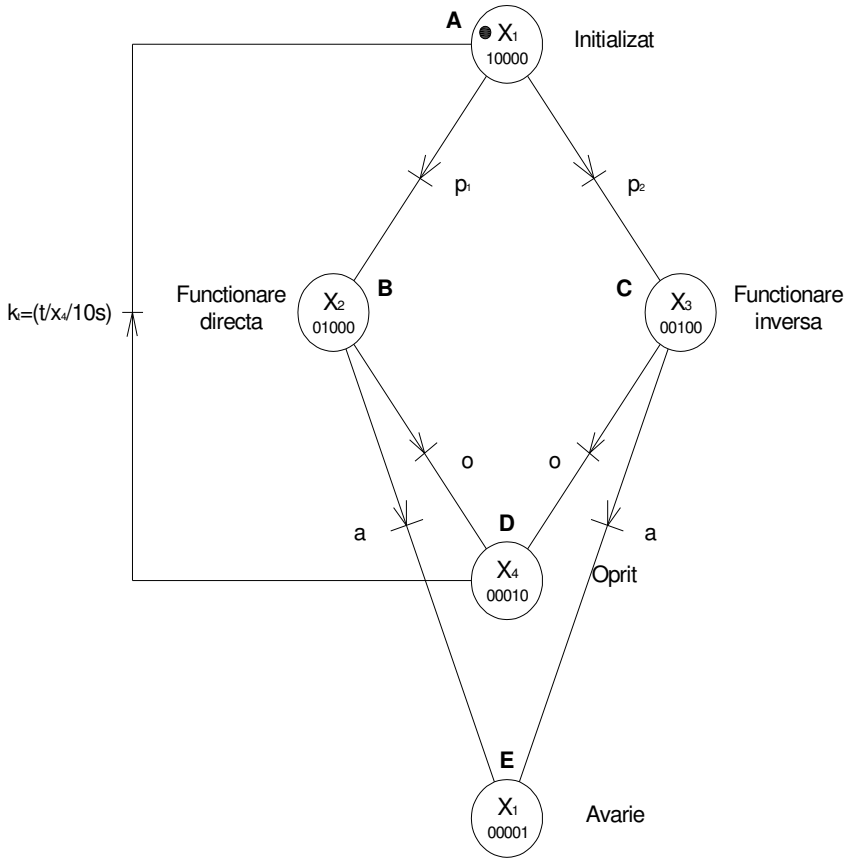


Fig. 4.95 Caietul de sarcini pentru automatul motorului reversibil

Caietul de sarcini sub formă de rețea Petri pentru automatul servomotorului reversibil ce inițializare este prezentat în Fig. 4.95.

Pe baza acestui caiet de sarcini vom realiza o rețea Petri conformă în care în care vom rezolva problema conflictelor şi a blocării ca în Fig. 4.96.

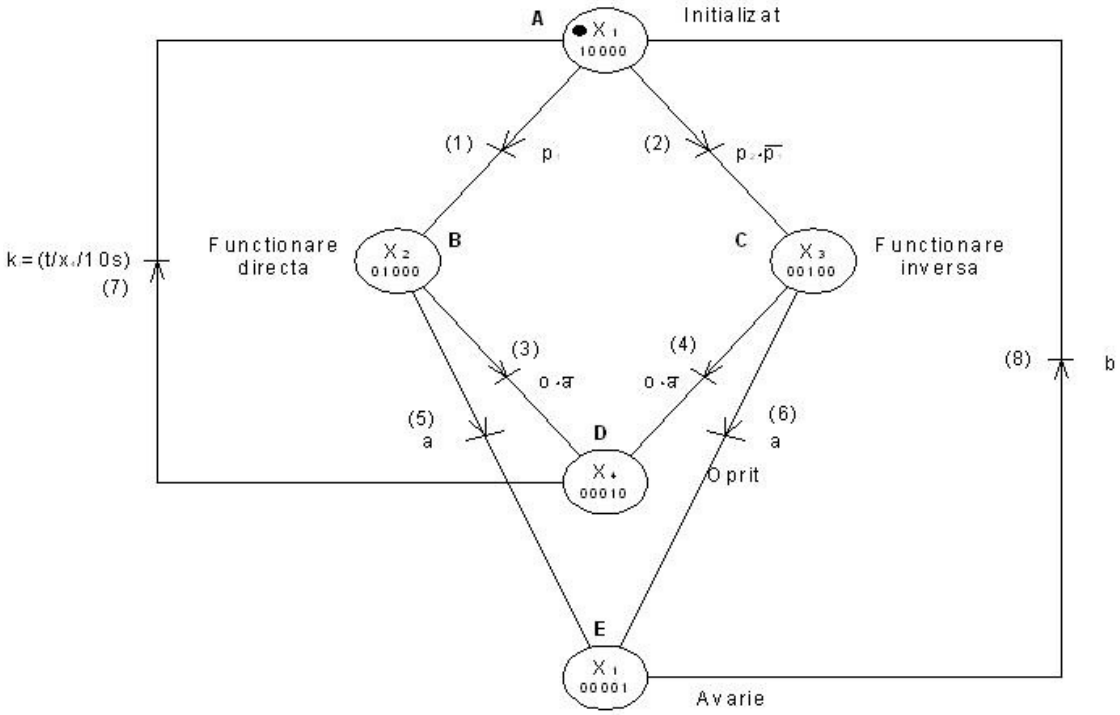


Fig. 4.96 Rețeaua Petri pentru automatul motorului reversibil

Se stabilesc ciclurile repetitive:

A – B – D – A

A – C – D – A

A – B – E – A

A – C – E – A

Conflictele structurale efective au fost eliminate prin introducerea de condiții de prioritate.

Calculul ecuațiilor va fi făcut cu ajutorul următoarelor formule:

Pentru etapele inițial active:

$$x_i^{t+\Delta} = s_i + \overline{r_i} \cdot x_i^t + i$$

Pentru etapele inițial inactive:

$$x_i^{t+\Delta} = \left(s_i + \overline{r_i} \cdot x_i^t \right) \cdot \bar{i}$$

Astfel ecuațiile pentru locațiile noastre vor fii:

Pentru locația A – reprezintă inițializarea servomotorului:

A → x1t :

Setarea bitului x1: $s_1^t = x_4 \cdot k_t + x_5 \cdot b$

Resetarea bitului x1: $r_1^t = x_1 \cdot p_1 + x_1 \cdot p_2 \cdot \overline{p_1}$

$$x_1^{t+\Delta} = x_4 \cdot k_t + x_5 \cdot b + \overline{p_1} \cdot \overline{p_2} \cdot x_1 + i$$

Pentru locația B – reprezintă pornirea directă a servomotorului:

B → x2t :

Setarea bitului x2: $s_2^t = p_1 \cdot x_1$

Resetarea bitului x2: $r_2^t = x_2 \cdot a + x_2 \cdot o \cdot \overline{a}$

$$x_2^{t+\Delta} = \left(p_1 \cdot x_1 + \overline{a} \cdot \overline{o} \cdot x_2 \right) \cdot \overline{i}$$

Pentru locația C – reprezintă pornirea inversă a servomotorului:

C → x3t :

Setarea bitului x3: $s_3^t = x_1 \cdot p_2 \cdot \overline{p_1}$

Resetarea bitului x3: $r_3^t = x_3 \cdot a + x_3 \cdot o \cdot \overline{a}$

$$x_3^{t+\Delta} = \left(x_1 \cdot p_2 \cdot \overline{p_1} + \overline{a} \cdot \overline{o} \cdot x_3 \right) \cdot \overline{i}$$

Pentru locația D – reprezintă oprirea servomotorului:

D → x_4^t :

Setarea bitului x_4: $s_4^t = x_2 \cdot o \cdot \overline{a} + x_3 \cdot o \cdot \overline{a} = \left(x_2 + x_3 \right) \cdot o \cdot \overline{a}$

Resetarea bitului x_4: $r_4^t = x_4 \cdot k_t$

$$x_4^{t+\Delta} = \left(\left(x_2 + x_3 \right) \cdot o \cdot \overline{a} + \overline{k_t} \cdot x_4 \right) \cdot \overline{i}$$

Pentru locația E – reprezintă avaria:

E → x_5^t :

Setarea bitului x_5: $s_5^t = (x_2 + x_3) \cdot a$

Resetarea bitului x_5: $r_5^t = x_5 \cdot b$

$$x_5^{t+\Delta} = \left((x_2 + x_3) \cdot a + \bar{b} \cdot x_5 \right) \cdot \bar{i}$$

Scrierea ecuațiilor pentru implementarea acestora cu contacte şi relee:

Pentru a putea implementa aceste ecuații, va fi necesar să facem următoarele substituții:

$x_n^{t+\Delta} \rightarrow K_n^{t+\Delta}$

$x_n^t \rightarrow k_n^t$

Astfel cele cinci ecuații corespunzătoare celor cinci locații din caietul de sarcini al servomotorului vor fi:

$$K_1^{t+\Delta} = k_4 \cdot k_t + k_5 \cdot b + \overline{p_1} \cdot \overline{p_2} \cdot k_1 + i$$
$$K_2^{t+\Delta} = \left(p_1 \cdot k_1 + \bar{a} \cdot \bar{o} \cdot k_2 \right) \cdot \bar{i}$$
$$K_3^{t+\Delta} = \left(k_1 \cdot p_2 \cdot \overline{p_1} + \bar{a} \cdot \bar{o} \cdot k_3 \right) \cdot \bar{i}$$
$$K_4^{t+\Delta} = \left((k_2 + k_3) \cdot o \cdot \bar{a} + \overline{k_t} \cdot k_4 \right) \cdot \bar{i}$$
$$K_5^{t+\Delta} = \left((k_2 + k_3) \cdot a + \bar{b} \cdot k_5 \right) \cdot \bar{i}$$

Pentru implementarea ecuațiilor servomotorului cu ajutorul unui EASY 621-DC-TC vom avea următoarele intrări:

p_1	--	I01
p_2	--	I02
i	--	I03
o	--	I04
a	--	I05
b	--	I06

Ieşirile vor fi:

K1 -- Q01
K2 -- Q02
K3 -- Q03
K4 -- Q04
K5 -- Q05

Programul a fost realizat cu ajutorul mediului de proiectare asistată de calculator EASY-SOFT 5.00 [Pro]. În cadrul acestui program am ales un EASY 621-DC-TC. Programarea automatului servomotorului s-a făcut sub forma de Ladder Diagram:

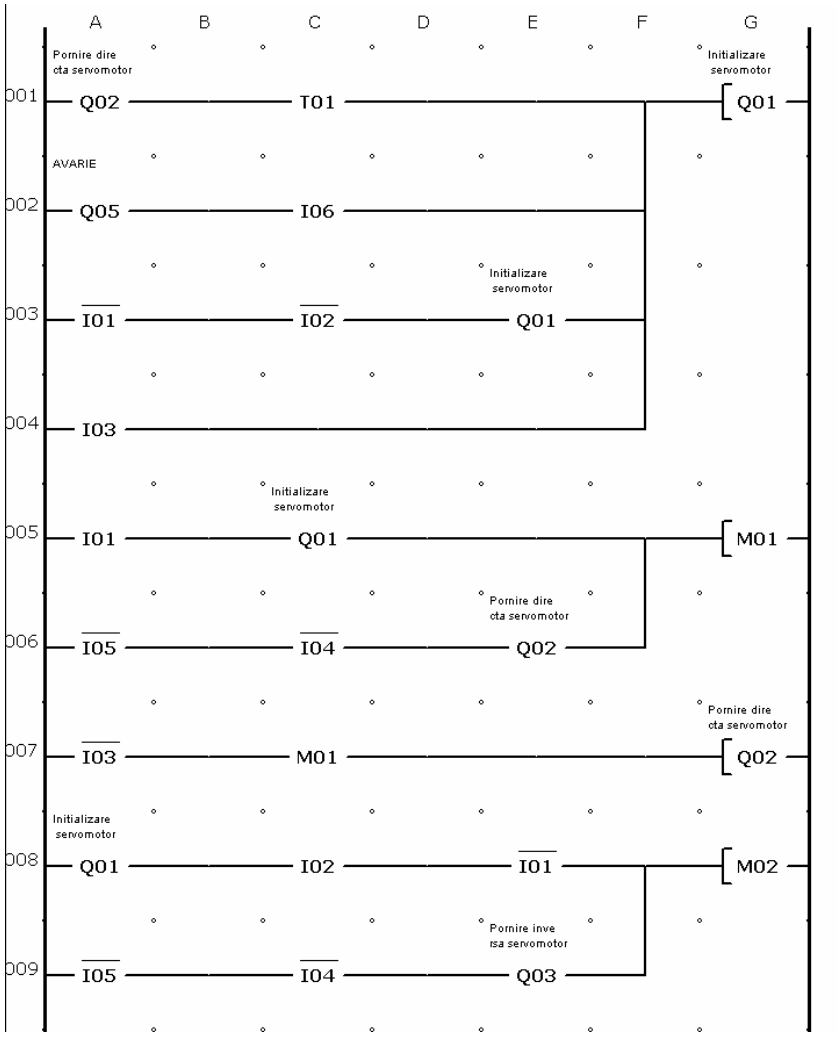


Fig. 4.97 Prima parte a programul automatuluuii pentru motorul reversibil

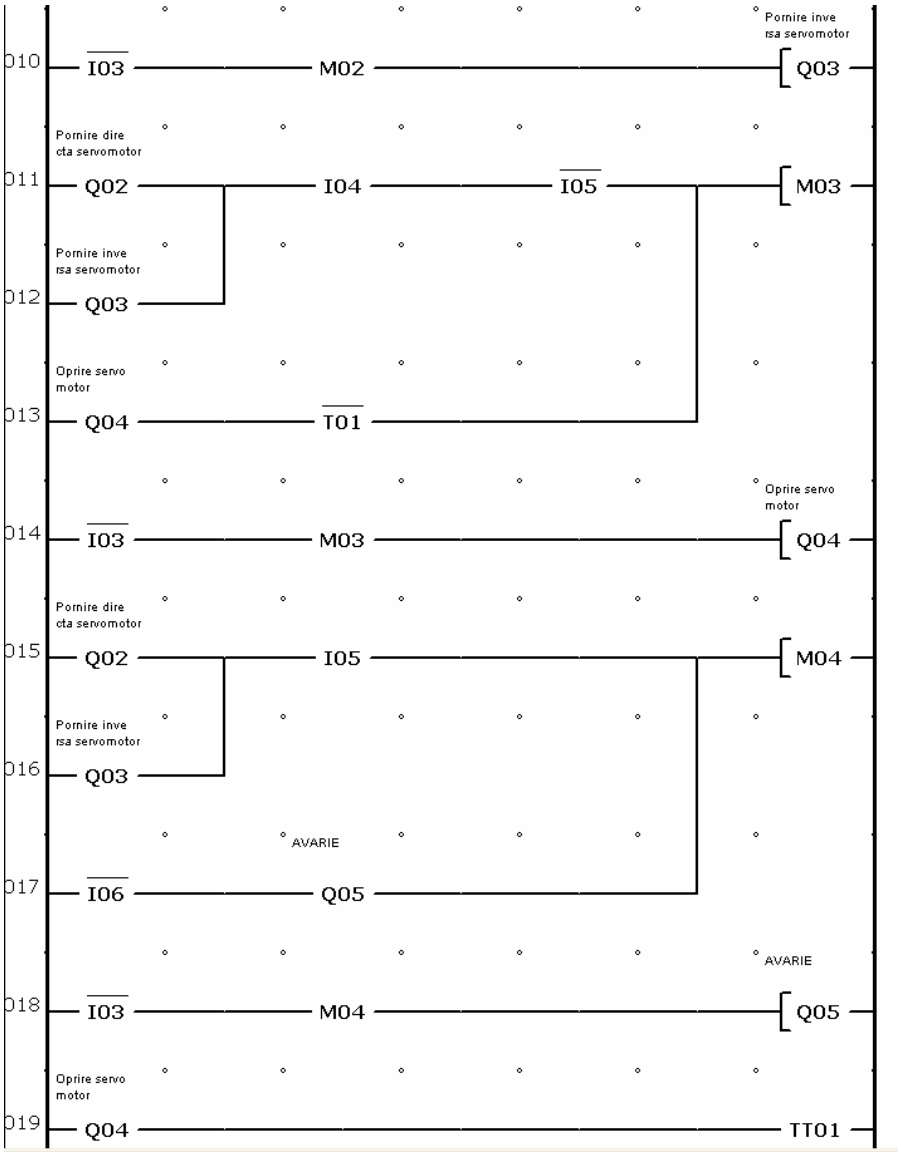


Fig. 4.98 A doua parte a programului pentru automatul motorului reversibil

Schema desfăşurată electrică a automatului programabil Easy Relay 621-DC-TC

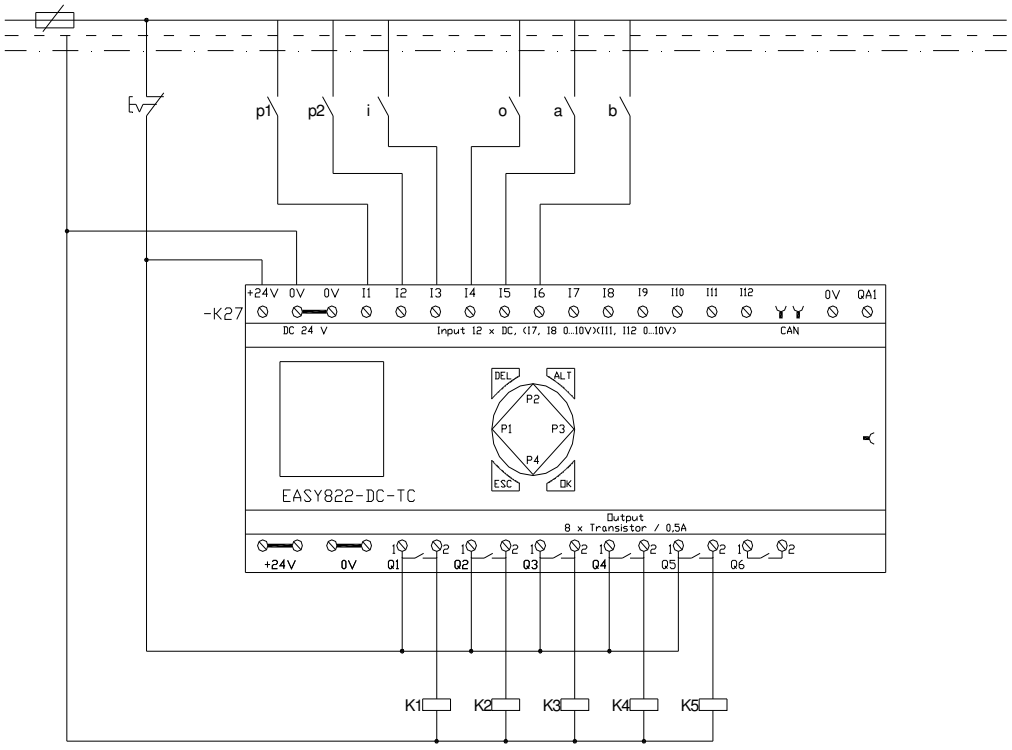


Fig. 4.99 Schema electrică des desfăşurată a automatului pentru motorul reversibil

4.20.4. Automat cu contacte şi relee pentru ascensor

Ascensorul trebuie să deservească corpul de producție al unei fabricii, o clădire P+2, pentru transportul materiei prime de la un nivel la altul.

Apelul poate fi făcut fie din interiorul cabinei, fie din exterior de la un anumit etaj. Ascensorul are în interior 3 butoane: e_1, e_2, e_3 corespunzătoare nivelelor parter, etaj 1şi etaj 2. Butoanele sunt de tip normal deschis cu revenire.

Daca ascensorul este la un nivel oarecare şi se apasă un buton corespunzător altui nivel, liftul se pune în mişcare în sus sau în jos. Dacă se apasă concomitent două sau trei butoane liftul se va pune în mişcare către etajul prioritar. Nivelul prioritar este parterul (e_1), urmat apoi de etajul 1 (e_2) şi mai apoi etajul 2 (e_2). Există la fiecare nivel un palpator selector etaj care opreşte cursa liftului.

Schema tehnologică a ascensorului:

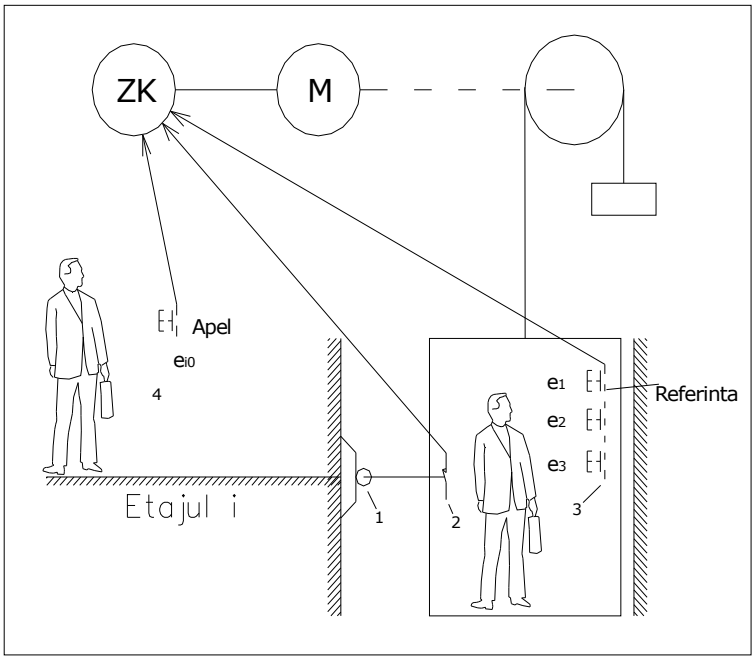


Fig. 4.100 Schema tehnologică cu echipamentul de automatizare pentru automatul ascensorului

Legendă:

1 – palpatorul selectorului de etaj;

2 – contact generator impulsuri (traductor poziție);

3 – referință din cabină;

4 – apel de pe palier.

Rețeaua Petri de tip grafcet pentru memorarea impulsurilor de referință (apel):

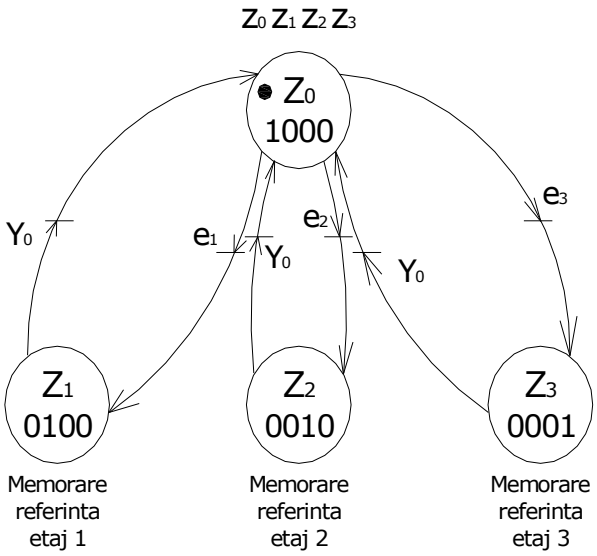


Fig. 4.101 Automat pentru memorarea impulsurilor de apel

Rețeaua Petri de tip grafcet pentru comanda motorului reversibil:

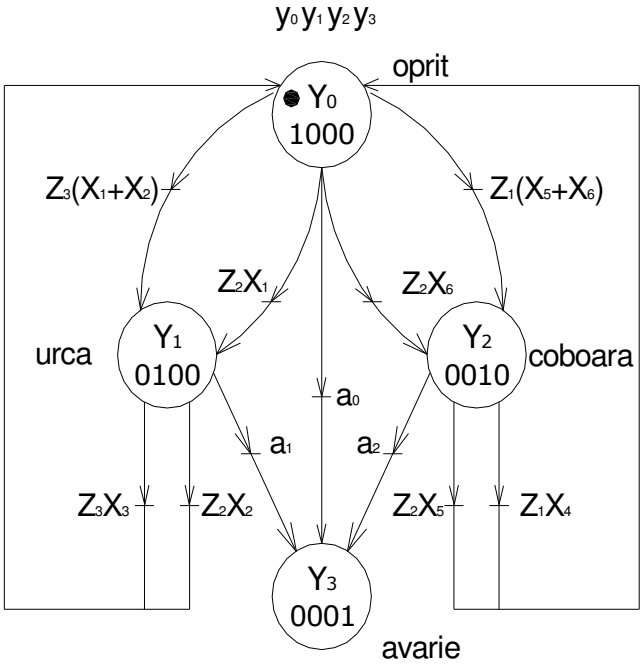


Fig. 4.102 Automat pentru comanda motorului reversibil

Rețeaua Petri de tip grafcet pentru selectoarele de etaj:

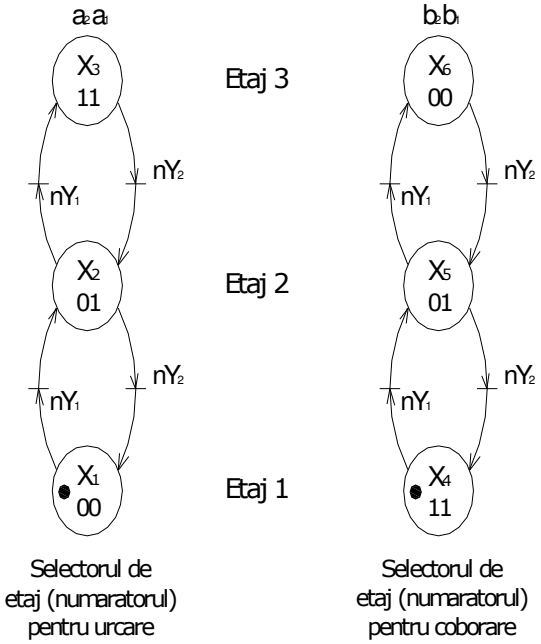


Fig. 4.103 Automat pentru selectorul de etaj

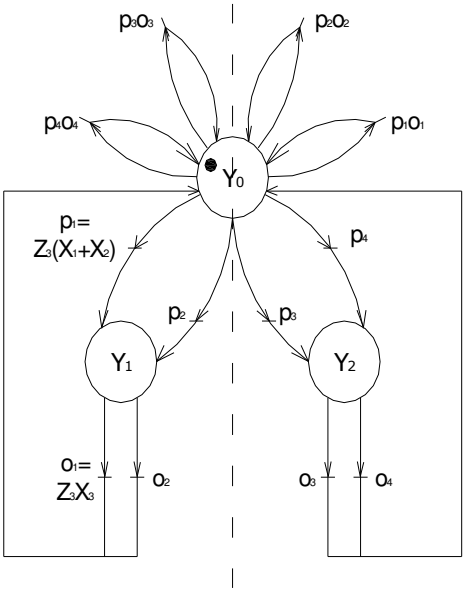


Fig. 4.104 Asigurarea priorităţii la oprire

Analiza reţelelor Petri (interpretarea, blocarea şi rezolvarea conflictelor). Simetrie şi repetare. Se tratează numai referinţele şi urcarea. Datorită interpretării între reţelele Petri şi restricţiilor tehnologice, ciclurile repetitive sunt într-un număr mult mai mic. De exemplu avem Y_0 p_1 Y_1 o_1 Y_0 şi nu Y_0 p_1 Y_1 o_2 Y_0.

La activarea simultană a două etape are prioritate Y_0. Apar arcele în buclă şi reţeaua Petri degenerează.

Multe conflicte structurale nu sunt efective datorită modului în care sunt construite selectoarele de urcare şi de coborâre. Se înlătură degenerarea prin interblocare, eliminând conflictele efective corespunzătoare.

Simetrie

Prioritate Y_0

Cicluri repetitive realizabile fizic

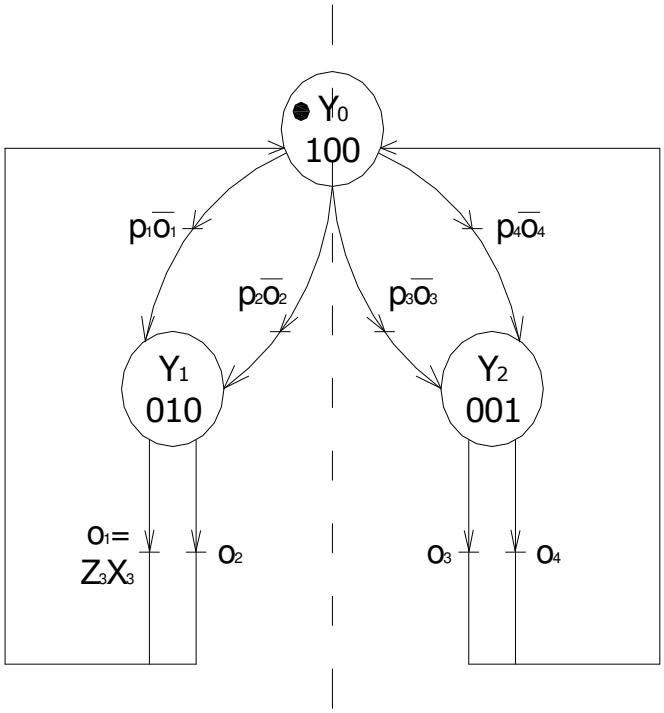


Fig. 4.105 Interblocarea

Calculul pentru determinarea relațiilor logice la implementarea cu contacte şi relee.
Pentru etapele inițial active:

$$x_i^{t+\Delta} = s_i + \overline{r_i} \cdot x_i^t + i$$

Pentru etapele inițial inactive:

$$x_i^{t+\Delta} = \left(s_i + \overline{r_i} \cdot x_i^t \right) \cdot \bar{i}$$

Pentru simplificarea calculelor se tratează pe rând, numai ciclurile realizabile fizic.
Calculul pentru determinarea relației logice corespunzătoare etapei Y_0.
Calea pentru ciclul repetitiv Y_0 p_1 Y_1 o_1 Y_0.

$$s_1 = p_1 \cdot \overline{o_1} \cdot y_0$$

$$r_1 = o_1 \cdot y_1$$

$$y_1^{t+\Delta} = \{ p_1 \cdot \overline{o_1} \cdot y_0 + \overline{o_1 \cdot y_1} \cdot y_1 \} \cdot \bar{i} =$$

$$= \{ p_1 \cdot \overline{o_1} \cdot y_0 + \left(\overline{o_1} + \overline{y_1} \right) \cdot y_1 \} \cdot \bar{i}$$

$$= \{ p_1 \cdot \overline{o_1} \cdot y_0 + \overline{o_1} \cdot y_1 + \overline{y_1} \cdot y_1 \} \cdot \bar{i}$$

$$= \{\overline{o_1} \cdot \left(p_1 \cdot y_0 + y_1 \right)\} \cdot \bar{i}$$

Numai la implementarea cu contacte şi relee se pot face următoarele simplificări: Se renunță la bitul de cod y_0. Deoarece inițializarea se poate obține în acest caz prin tăierea alimentării se renunță la i. Deoarece toate evenimentele sunt memorate se renunță la automenținere. Deci,

$$y_1^{t+\Delta} = \overline{o_1} \cdot p_1$$

Asemănător şi pentru calea Y_0 p_2 Y_1 o_2 Y_0 :

$$y_1^{t+\Delta} = \overline{o_2} \cdot p_2$$

Deci, $y_1^{t+\Delta}$ devine pentru ambele cazuri:

$$y_1^{t+\Delta} = \overline{o_1} \cdot p_1 + \overline{o_2} \cdot p_2$$

Înlocuind evenimentele p_1 , p_2 , o_1 , o_2 cu valorile lor se obține:

$$y_1^{t+\Delta} = \left(\overline{Z_3 \cdot X_3}\right) \cdot Z_3 \cdot \left(X_1 + X_2\right) + \left(\overline{Z_2 \cdot X_2}\right) \cdot Z_2 \cdot X_1 =$$

$$= \left(\overline{Z_3} + \overline{X_3}\right) \cdot Z_3 \cdot \left(X_1 + X_2\right) + \left(\overline{Z_2} + \overline{X_2}\right) \cdot Z_2 \cdot X_1 =$$

$$= \left(\overline{Z_3} \cdot Z_3 + \overline{X_3} \cdot Z_3\right) \cdot \left(X_1 + X_2\right) + \left(\overline{Z_2} \cdot Z_2 + \overline{X_2} \cdot Z_2\right) \cdot X_1 =$$

$$= \overline{X_3} \cdot Z_3 \cdot \left(X_1 + X_2\right) + \overline{X_2} \cdot Z_2 \cdot X_1 =$$

Se înlocuiesc variabilele etapelor X_1 , X_2 , X_3 , Z_2 , Z_3 cu biții care îi caracterizează:

$$y_1^{t+\Delta} = \left(\overline{a_2 \cdot a_1}\right) \cdot z_3 \cdot \left(\overline{a_2} \cdot \overline{a_1} + \overline{a_2} \cdot a_1\right) + \left(\overline{\overline{a_2} \cdot a_1}\right) \cdot z_2 \cdot \overline{a_2} \cdot \overline{a_1} =$$

$$= \left(\overline{a_2} + \overline{a_1}\right) \cdot z_3 \cdot \overline{a_2} \cdot \left(\overline{a_1} + a_1\right) + \left(a_2 + \overline{a_1}\right) \cdot z_2 \cdot \overline{a_2} \cdot \overline{a_1} =$$

$$= \left(\overline{a_2} + \overline{a_1}\right) \cdot z_3 \cdot \overline{a_2} + \left(a_2 + \overline{a_1}\right) \cdot z_2 \cdot \overline{a_2} \cdot \overline{a_1} =$$

$$= \left(\overline{a_2} + \overline{a_1}\right) \cdot z_3 \cdot \overline{a_2} + \overline{a_1} \cdot \overline{a_2} \cdot z_2 =$$

$$= \overline{a_2} \cdot \left(1 + \overline{a_1}\right) \cdot z_3 + \overline{a_1} \cdot \overline{a_2} \cdot z_2 =$$

$$= \overline{a_2} \cdot z_3 + \overline{a_1} \cdot \overline{a_2} \cdot z_2 =$$

$$= \overline{a_2} \cdot \left(z_3 + \overline{a_1} \cdot z_2\right)$$

Codificare specială pentru implementarea cu contacte şi relee. Se renunță şi la butonul de inițializare i.

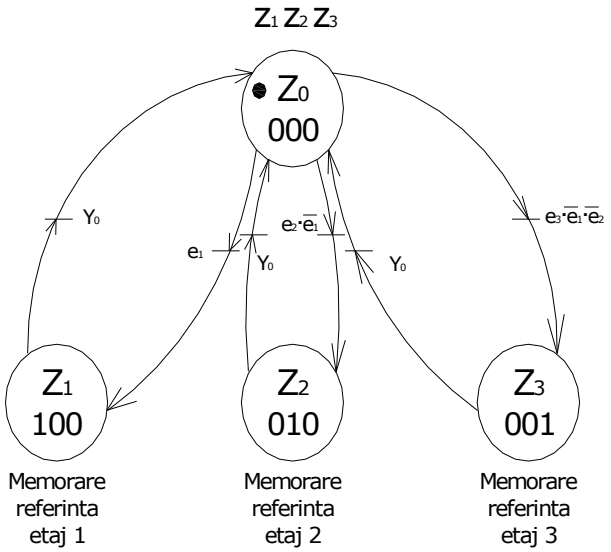


Fig. 4.106 Interblocarea pentru automatul de memorare al etajului

Schema clasică în punte cu contacte şi relee.

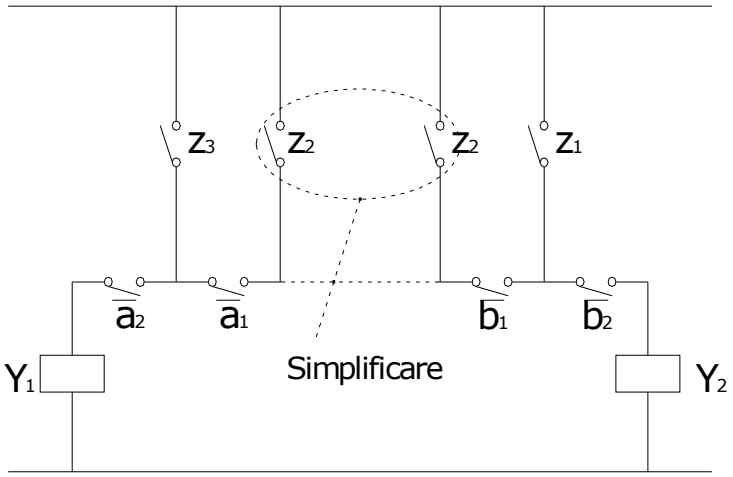


Fig. 4.107 Schema electrică desfăşurată a automatului cu contacte şi relee pentru ascensor

4.20.5. Automat cu APL pentru anclanşarea automată a rezervei tehnologice

Schema tehnologică cu echipamentul de automatizare pentru o instalație formată dintr-un rezervor şi două pompe este prezentată în Fig. 4.108. Scopul automatizării îl constituie păstrarea nivelului la o valoare cuprinsă între un minim şi maxim chiar şi în condițiile avarierii uneia dintre cele două pompe. Pentru aceasta se foloseşte traductorul de nivel 1, automatul programabil 2 şi elementul de execuție 3. Se specifică intrările şi ieşirile din automatul programabil, care au următoarea semnificație:

- I1 – intrarea 1 formată din contacul normal deschis al unui buton în impuls pentru selectarea pompei P1 pentru regimul de lucru şi al pompei P2 pentru regimul de rezervă tehnologică.
- I2 – intrarea 2 formată din contactul normal deschis al unui buton în impuls pentru selectarea pompei P2 pentru regimul de lucru şi al pompei P1 pentru regimul de rezervă tehnologică.
- I3, I4 – intrările 3 şi 4 formate din contactele normal închise ale traductoarelor de avarie ale pompelor P1, respectiv P2. La apariția avariei contactele se deschid. În felul acesta se asigură funcționarea în regim de *fail safe.*
- I5 – intrarea 5, un contact normal închis în impuls, pentru inițializarea automatului.
- I6 – intzrarea 5, formată dintr-un contact normal deschis, în impuls, pentru oprirea pompelor.
- I7 – intrarea 7, formată dintr-un contact normal deschis, în impuls cu cheie, pentru confirmarea avariei.
- Q1, Q2 – ieşirile 1 şi 2, formate din contacte normal deschise, pentru acționarea şi semnalizarea acționării pompelor 1 şi 2.
- Q3, Q4 – ieşirile 3 şi 4, formate din contacte normal deschise, pentru semnalizarea prin pâlpâire a avariei pompelor 1 şi 2. După acționarea butonului de oprire de la intrarea I6 pâlpâirea încentează, dar lămpile semnalizează în continuare avaria până la confirmarea ei cu butonul I7.
- Q5, Q6 – ieşirile 5 şi 6, formate din contacte normal deschise, pentru semnalizarea selectării pompei P1, respectiv P2, pentru funcționarea în regim de

lucru.

Algoritmul de funcționare al automatului este descris de rețeaua Petri din Fig. 4.109. După pornirea APL locația L10 este activă. Acționarea butonului de inițializare i5 activează locația L11. Din această stare se poate selecta regimul de lucru al pompelor Q1 şi Q2 care poate fi de lucru sau de rezervă. Rețeaua este simetrică şi din această cauză în continuare se va analiza doar situația în care se alege pompa Q1 de lucru şi se activează locatia L12, situatie semnalizată de lampa Q5. Nivelul din rezervor poate lua trei valori.. Dacă este mai mic decăt

min1=2,9 atunci a1=1, se activează L13 în care porneşte pompa Q1. Nivelul va creşte şi atunci când va depăşi Max=3 evenimentul a3=1 şi pompa Q1=0 se opreşte deoarece se activează din nou L12. Acesta este ciclul repetitiv principal de funcţionare. Dacă apare o avarie funcţională[167] sau o creştere de sarcină atunci pompa Q1 nu face faţă şi nivelul continuă să scadă. Atunci când este nivelul mai mic decât min2=2 un timp mai lung decât d4 apare evenimentul a2.d4=1, se activează L14 şi porneşte pompa de rezervă Q2=1. Când acţiunea este încununată de succes nivelul creşte, depăşeşte valoarea maximă şi apare din nou evenimentul a3. Se revine în L1 unde ambele pompe sunt oprite. Din L13 care reflectă regimul de funcţionare normală cu pompa de lucru Q1 se poate trece ăn starea de avarie L18 atunci când se deschide contactul i3.

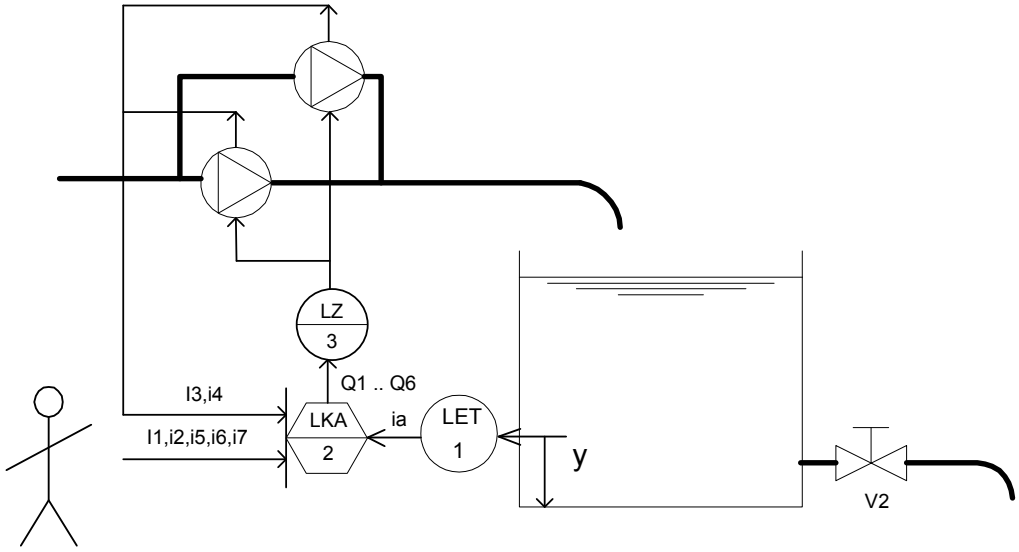


Fig. 4.108 Schema tehnologică cu echipamentul de automatizare pentru anclanşarea automată a rezervei tehnologice

În locaţia L18 funcţionează pompa Q2=1 şi semnalizează intermitent avaria Q3≈1. Avarierea pompei a doua conduce la oprirea ei Q2=0 în locaţia L20 Lămpile Q3 şi Q4 pâlpâie. Dacă se apasă butonul de oprire i6 din regimul de funcţionare normală L11 cu ambele pompe oprite şi cu posibilitatea de alegere a pompei de lucru şi a pompei de rezervă. Acţionarea butonului de oprire din starea de avarie activea activează una din locaţiile L21, L22 sau L23 în care lîmpile de avarie Q3 şi Q4 nu mai pâlpâie. Confirmarea avariei prin apăsarea butonului cu cheie i7 readuce automatul în starea iniţializată L11.

La început se face analiza performanţelor reţelei. Prin analiza structurii observâ imediat că este simetrică şi nereversibilă în locaţia L10.

[167] In cadrul unei avarii funcţionale echipamentul nu este scos din funcţiune ci lucrează cu performanţe reduse.

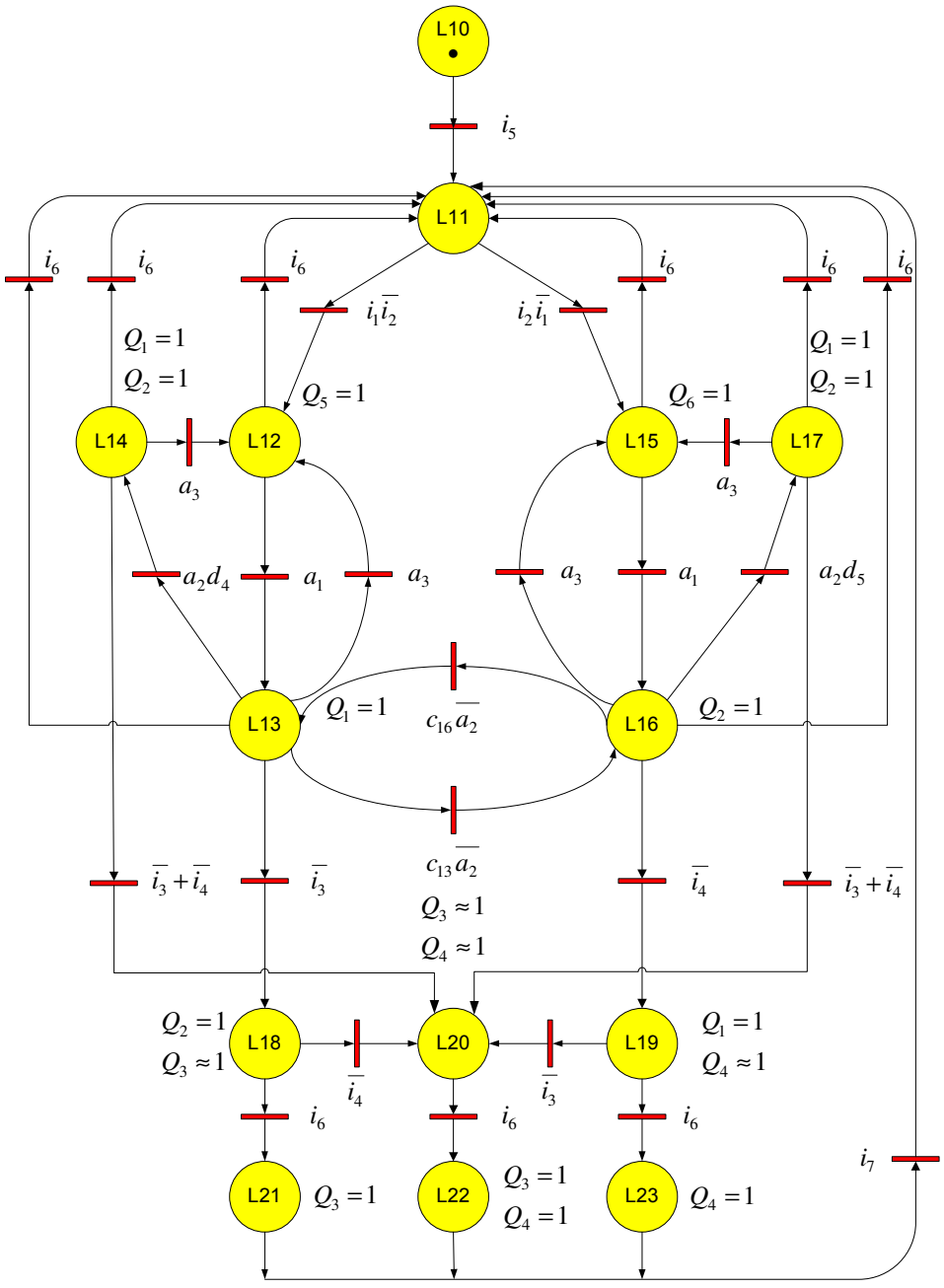


Fig. 4.109 Caietul de sarcini sub formă de rețea Petri pentru anclanşarea automată a rezervei tehnologice

Reversibilitatea se asigură observând ca se poate ajunge în L10 din oricare altă locație sau la pornire prin acționarea butoanelor de pornire şi oprire ale automatului.

Analiza structurală pune în evidenţî foarte multe conflicte structurale şi efective. Examinând consecinţele acestor conflicte s-a făcut interblocarea numai pentru L11, L13 şi L14 pentru a nu mării foarte mult complexitatea algoritmului.

În urma analizei comportamentale se observă că reţeaua nu se blochează, deci are performanţa de viabilitate şi nu are niciodată mai mult de o marcă în fiecare locaţie, deci satisface şi performanţa de siguranţă. Reţeaua Petri este deci interpretată, sigură, viabilă şi fără conflicte efective. Prin urmare este numită conformă şi coincide cu o clasă largă de grafceturi frecvent utilizate în practică. Se poate aplica una dintre metodele de proiectare prezentate în paragrafele precedente. Alegem implementarea cu APL şi metoda de programare a schemei desfăşurate cu folosirea automatelor elementare RS cu prioritate la pornire. În acest caz este necesară numai codificarea locaţiilor reţelei şi determinarea condiţiilor de setare şi resetare pentru fiecare locaţie.

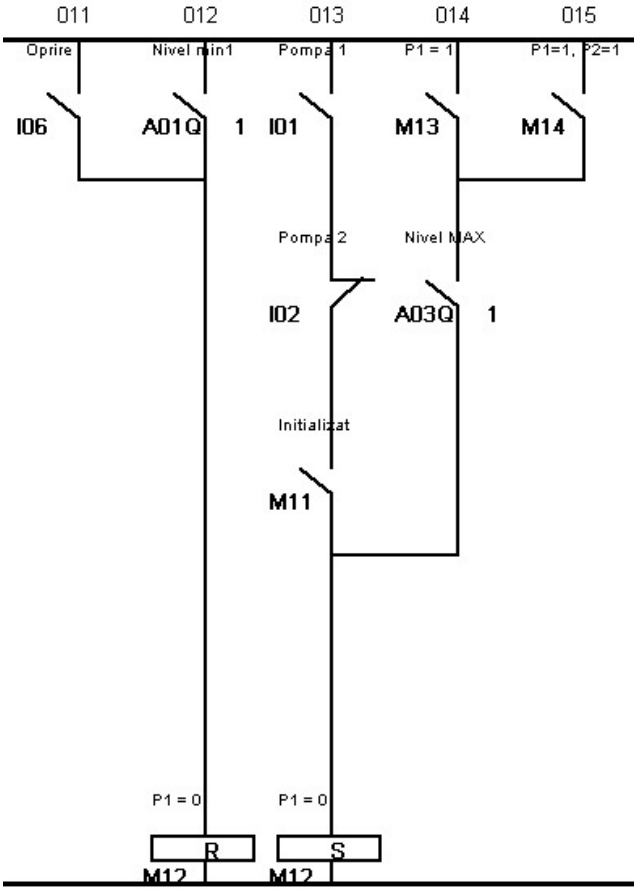


Fig. 4.110 Programul pentru locaţia L12

Locaţiile Reţelei Petri se codifică distibutiv ca în Tab. 4.10. Fiecărui bit îi corespunde un automat elementar (bistabil) tip RS cu prioritate la setare (pornire). Pentru APL Easy Relay aceste automate iau forma unor markeri tip RS. Markerul are o condiţie de resetare şi una de setare care rezultă din reţeaua Petri. Deoarece reţeaua Petri este de tip maşina de stare pentru

care la fiecare tranziție intră şi iese un singur arc, condiția de resetare este formată numai din evenimentele ataşate tranzițiilor posterioare locației L12.

Tab. 4.10 Codificarea locațiilor rețelei Petri pentru AAR

Locația	***Codul***	***Automatul elementar***
L10	000000000000000000000000	
L11	100000000000000000000000	M11
L12	010000000000000000000000	M12
L13	001000000000000000000000	M13
.....	.....	.....
L21	000000000000000000000100	M21
L22	000000000000000000000010	M22
L23	000000000000000000000001	M23

Relația logică pentr condiția de resetare se determină observând că sunt numai două arce care ies din L12 condiționate de tranzițiile cu evenimentele i6 şi a1.

$$RM12 = i_6 + a_1$$

Relația logică pentru condiția de setare a bitului corespunzător L12 rezultă tot din rețeaua Petri prin examinarea arcelor care intră în ea.

$$SM12 = M11 i_1 \overline{i_2} + M13 a_3 + M14 a_3 = M11 i_1 \overline{i_2} + (M13 + M14) a_3$$

Programul sub formă de schemă desfăşurată pentru markerul M12 corespunzător locației L12 este prezentat în Fig. 4.110. Spre deosebire de condiția de resetare aici fiecare eveniment care declanşează tranziția este validat de activarea locației precedente corespunzătoare.

Programul pentru celelalte locații se realizează în mod asemănător.

5. Sisteme de conducere în caz de pericol

Cauzele care provoacă pericole în clădiri, grupuri de clădiri şi locuinţe sunt:

- Incendiul;
- Efracţia;
- Jaful;
- Avariile instalaţiilor tehnologice;
- Catastrofe naturale;
- Abateri de la regulile de convieţuire.

Echipamente clasice de conducere pentru aceste pericole sunt sistemele de securitate la efracţie şi jaf, sistemele de control al accesului şi sistemele de siguranţă la foc. Aceste sisteme sunt construite în jurul unor automate cu evenimente discrete care reacţionează la semnalele transmise de traductori specifici şi execută diferite acţiuni care să prevină pericolele şi să limiteze pierderile.

Sistemele de conducere automată în caz de pericol trebuie să realizeze următoarele acţiuni:

- ♦ Anticiparea prezenţei pericolului;
- ♦ Semnalizarea acustică şi vizuală (alarmarea) la nivel local, la nivel central şi la distanţă;
- ♦ Protecţie automată pentru limitarea pierderilor;
- ♦ Documentarea alarmelor în sensul explicării gradului de pericol, a locului în care au apărut şi a acţiunilor recomandate operatorului uman.
- ♦ Managementul alarmelor[168].

Pericolele se definesc în funcţie de scopul protecţiei:

- ✓ Viaţa;
- ✓ Bunuri materiale;
- ✓ Mediul natural înconjurător;
- ✓ Mediul social de convieţuire.

Standardele EN54 şi CEI 839 precizează că sistemul de conducere automată în cazul pericolelor se compune din:

1. Centrala ce conducere. Adeseori aceasta este desemnată sub denumirea de echipament de detecţie şi semnalizare sau centrală de alarmă. După cum am menţionat echipamentele moderne pot avea şi alte funcţiuni în afară de alarmare, cum ar fi protecţia sau documentarea.
2. Butoane de avertizare manuală.

[168] Gestion des alarmes în franceză, Alarm management în engleză.

3. Traductoarele de pericole. Aceste pot fi de tipul detectoarelor de avarie, a traductoarelor (monitoare analogice), sau a traductoarelor inteligente. Aceste dispozitive formează semnalele de intrare în centrală.
4. Dispozitive de alarmare, de protecție automată sau de recepție a semnalelor emise de centrală. Aceste dispozitive sunt conectate la ieşirea centralei.
5. Sursa de alimentare.

Sistemul de conducere în caz de pericol trebuie să respecte următoarele principii în funcționarea sa:

a) Timpul de acționare al sistemului de conducere trebuie să fie mai mic decât timpul minim în care sistemul protejat rezistă la pericol.
b) Fiabilitatea sistemului de conducere trebuie să fie mai mare decât a sistemului protejat.
c) Subordonarea sistemului de protejat față de sistemul de conducere trebuie asigurată numai în caz de necesitate. Sistemul de conducere trebuie să aibă capacitatea de a acționa corespunzător limitării pierderilor numai în cazuri reale de necesitate. Nu este permisă această acționare datorită unor efecte laterale a interacțiunii cu sistemul protejat, a defectării acestuia, sau a apariției unor defecte în însuşi sistemul de conducere.
d) Nefuncționarea sistemului de conducere nu trebuie să împiedice funcționarea sistemului protejat
e) Sistemul de conducere este subordonat operatorului uman. singurul care poate interpreta situații complexe periculoase şi care poate lua deciziile necesare.

Funcționarea unui sistem de conducere în caz de pericol cuprinde mai multe etape distincte:

A. Prima şi cea mai importantă etapă constă în detecția pericolelor. În această etapă sistemul de conducere nu trebuie să influențeze în nici un fel activitatea sistemului protejat.
B. Identificarea pericolului şi verificarea identificării. Declanşarea alarmei. În această etapă operatorul uman poate influența decisiv evoluția ulterioară a ansamblului sistem de conducere - sistem – protejat. Din această stare se poate trece imediat automat sau la intervenția operatorului în starea următoare C de urgență – protejare. Dacă alarmă nu a fost reală se trece în prima stare A.
C. Etapa de urgență în care se execută toate acțiunile necesare limitării pierderilor datorate evoluției pericolului. Dacă pericolul persistă sistemul de conducere poate utiliza metode de supraviețuire prin restructurare şi poate subordona sistemul protejat. La terminarea pericolului se revine în etapa A.

Modul specific de funcționare al sistemelor de conducere în caz de pericol a impus ca integrarea acestora în sistemele de domotică să se realizeze numai la nivelul superior de management. În felul acesta se respectă principiul subordonării față de operatorul uman şi se

poate asigura prioritatea maximă a evenimentelor de pericol față de alte evenimente provocate de avarii tehnologice şi defecțiuni ale sistemului de conducere şi liniilor de transmisie[169].

5.1. Conceptele sistemelor de securitate la efracție şi siguranță la foc

Terminologia folosită la sistemele de securitate la efracție şi jaf şi sistemele de siguranță la foc este puțin diferită de cea folosită la sistemele de conducere automată. Deşi există standarde în domeniu totuşi persistă unele neclarități datorită vitezei mari de dezvoltare a echipamentelor de acest tip şi a numărului mare de firme implicate în comercializarea şi construirea acestor sisteme de conducere. În continuare se prezintă câțiva termeni şi sensul lor din această lucrare.

- ✓ ACOPERIREA DETECTORULUI - Distanta maxima recomandata intre doua detectoare adiacente sau aria pe care un detector este proiectat sa o protejeze.
- ✓ ADRESABILITATE – Caracteristica dispozitivelor conectate la intrarea sau ieşirea unei centrale care permite identificarea lor pe baza unei adrese proprii prestabilite.
- ✓ ALARMA FALSA (NEDORITA) – Alarma având drept cauza evenimente zilnice ca gătitul, fumul de țigara, praf, insecte etc.
- ✓ AVERTIZARE - O indicație optica şi/sau acustica.
- ✓ CARACTERISTICA DE VERIFICARE A ALARMEI - Caracteristica a unui sistem de detecție automata a incendiului si a sistemelor de alarma gândita pentru a reduce alarmele false. Pentru ca un semnal sa fie acceptat ca alarma valida detectoarele de fum vor trebui sa raporteze condiție de alarma pe o perioada minima de timp sau vor trebui sa confirme condiția de alarma intr-o perioada de timp data, după inițializare.
- ✓ CENTRALĂ – Componentă a unui sistem de conducere în caz de pericol. Funcțiile principale ale centralei sunt: a) alimentarea cu energie a detectoarelor, b) managementul alarmelor, c) transmiterea la distanță a alarmelor şi c) supravegherea funcționării corecte a sistemului.
- ✓ CUPLOR RADIO FARA FIRE - Dispozitiv care recepționează, verifica si retransmite in cod binar, pe frecventa radio de putere mica, semnalele de alarma si semnalele de supervizare generate de detectoarele de fum si dispozitivele de inițiere.
- ✓ DEFECT – Stare a unui dispozitiv în care acesta nu-şi mai poate exercita funcțiile.
- ✓ DETECTOR – Traductor având cel puțin un element de tip comparator în partea sa finală de prelucrare a informației.
- ✓ DETECTOR (DE FUM) CU FASCICOL PROIECTAT - Principiul de funcționare al unui astfel de detector este monitorizarea luminii transmise de o sursa către un senzor fotosensibil. Când particulele de fum ajung in calea luminii o parte din

[169] Protocoalele de comunicație standardizate pentru clădiri şi locuințe cum ar fi BacNet sau EIB nu sunt potrivite pentru sistemele de conducere în caz de pericol.

aceasta este difuzata si o parte este absorbita, reducându-se astfel cantitatea de lumina care ajunge la senzor. Reducerea luminii intr-o anumita măsura, predeterminata, determina răspunsul detectorului.

- ✓ DETECTOR CU ESANTIONAREA AERULUI - Un astfel de detector consta intr-un sistem de țevi sau tuburi de distribuție, de la detector către aria de protejat. O pompa de are trage aer din zona supravegheata prin sistemul de țevi, către detector unde aerul este analizat.
- ✓ DETECTOR DE FLACARA - Dispozitiv care detectează radiația vizibila, in infra - roşu sau ultra - violet, produsa de foc.
- ✓ DETECTOR DE FUM - Dispozitiv care detectează particule de combustie vizibile sau invizibile.
- ✓ DETECTOR DE FUM ADRESABIL - Detectoare de fum care împreuna cu indicațiile de alarma si de avarie comunica unității centrale de control o adresa unica de identificare.
- ✓ DETECTOR DE FUM ANALOGIC* - Detector de fum, capabil sa comunice informații despre condițiile fumului si despre localizarea detectorului. Uzual, acest tip de detector comunica o adresa unica de identificare, împreuna cu un semnal analogic, care indica nivelul fumului, la locația respectiva (* in limbaj curent acest detector se mai numeşte detector adresabil).
- ✓ DETECTOR DE FUM CU IONIZARE - Acest tip de detector conține o mica cantitate de material radioactiv, ce ionizează aerul in camera de detecție, care devine conductor si permite curgerea curentului intre doi electrozi încărcați. In acest fel camera de detecție capătă o conductanța electrica efectiva. La intrarea particulelor de fum in zona ionizata conductanța scade prin ataşarea acestora la ioni, acest lucru generând scăderea mobilității ionilor. In acest fel, detectorul reacționează, daca conductanța este sub nivelul predeterminat.
- ✓ DETECTOR DE FUM FARA FIRE - Detector de fum care conține o baterie interna sau baterii care alimentează atât detectorul de fum cat si emițătorul radio. Sursa interna de alimentare este permanent supervizata si orice degradare a sa este comunicata unității centrale.
- ✓ DETECTOR DE FUM FOTOELECTRIC - Un astfel de detector de fum funcționează pe principiul luminii difuzate. El conține o sursa de lumina si un senzor fotosensibil, care sunt astfel aranjate încât razele emise de sursa de lumina nu cad direct pe senzorul fotosensibil. Când particulele de fum ajung in calea luminii, o parte din aceasta este difuzata către senzor, prin reflexie sau refracție, determinând răspunsul detectorului.
- ✓ DETECTOR DE FUM PE 2 FIRE - Detector de fum care generează condiția de alarma pe aceleaşi doua fire pe care este alimentat.
- ✓ DETECTOR DE FUM PE 4 FIRE - Detector de fum care generează condiția de alarma pe doua fire separate (bucla de inițiere) altele decât cele doua fire de alimentare.

- ✓ DETECTOR DE GRADIENT DE TEMPERATURA - Dispozitiv care semnalizează când temperatura creste cu o rata care depăşeşte o valoare predeterminata.
- ✓ DETECTOR DE PERICOL – Componentă a unui sistem de conducere în caz de pericol care cuprinde un senzor şi un element cu o caracteristică tip histerezis. Când parametrul supravegheat depăşeşte un anumit prag se transmite un semnal de alarmă la centrală.
- ✓ DETECTOR DE TEMPERATURA - Dispozitiv care detectează temperaturi anormal de înalte sau variaţii bruşte ale temperaturii in sensul creşterii acesteia.
- ✓ DETECTOR FOCALIZAT - Dispozitiv al cărui element de detecţie este concentrat pe o locaţie anume. Ca exemple tipice sunt detectoarele cu bimetal, detectoarele cu aliaj fuzibil, anumite detectoare de gradient, anumite detectoare de fum anumite detectoare termoelectrice.
- ✓ DETECŢIE A INCENDIULUI – Recunoaşterea timpurie a incendiului cu ajutorul unor traductoare inteligente de incendiu conectate pe o buclă analogică tolerantă la scurtcircuit şi întreruperi.
- ✓ DISPOZITIV DE INITIERE - Orice echipament, operat manual sau automat, care activat iniţiază o alarma printr-un dispozitiv de semnalizare.
- ✓ CENTRALĂ – Componentă a unui sistem de conducere în caz de pericol care recepţionează semnalele de alarmă şi defect şi ececută acţiunile prestabilite de semnalizare şi protecţie automată.
- ✓ EROARE – Situaţie (stare) nedorită apărută în prelucrarea unei informaţii.
- ✓ FILTRAREA UNUI SEMNAL – Procedeu de eliminare a influenţelor nedorite asupra unui semnal.
- ✓ FUZZY – Vag.
- ✓ HARDWARE – Construcţie fizică.
- ✓ INSTALAŢIE – Sistem.
- ✓ MONITOR ANALOGIC – Traductor analogic.
- ✓ PERICOL POSIBIL – Conjunctură nefavorabilă a unor parametrii de natură spaţio – temporală, electrică, mecanică, ambietală, etc. care poate determina rezultate nedorite faţă de procedura de funcţionare normală.
- ✓ PERICOL DECLANŞAT – Manifestarea rezultatelor nedorite ca urmare a prezenţei pericolului potenţial.
- ✓ PERTURBAŢIE – Cauză care provoacă modificarea semnalului de ieşire a sistemului.
- ✓ PRAG – Nivel (valoare) de referinţă.
- ✓ PROTOCOL – Set de reguli, constructive şi de procedură (software şi hardware) care reglementează schimbul de informaţii.
- ✓ RELEU CAPAT DE LINIE - Dispozitiv folosit pentru supervizarea alimentarii (uzual pentru detectoare de fum pe 4 fire) care se instalează in interiorul sau lângă ultimul dispozitiv din bucla.

- ✓ REȚEA DE COMUNICAȚIE – Suport fizic pe care mai mulți participanți pot schimba informații.
- ✓ SEMNAL DE ALARMA - Semnal ce indica o urgenta care cere intervenție imediata. Semnalul poate fi o alarma de foc de la un buton de avertizare manuala, o alarma de inundație, o alarma de la un detector de fum sau orice alt semnal care solicita o urgenta.
- ✓ SENZOR – Parte a unui dispozitiv care captează o schimbare de stare a parametrului spravegheat.
- ✓ SISTEM DE AVERTIZARE AUTOMATA LA INCENDIU - Sistem care consta in unități de control, dispozitive de inițiere si semnale de alarma, in care toate sau o parte din circuitele de inițiere sunt activate de dispozitive automate, cum ar fi detectoarele de fum.
- ✓ SISTEM DE CONDUCERE ÎN CAZ DE PERICOL – o mulțime de echipamente interconectate în scopul detectării pericolelor cât mai curând posibil şi a realizării acțiunilor de conducere (automate şi neautomate) necesare. Aceste acțiuni sunt de exemplu: declanşarea soneriei, transmisia alarmei la un centru de intervenție, comanda echipamentelor de protecție în caz de pericol, etc. Alarma se dă sub formă de semnalizare acustică şi optică în zona supravegheată sau în imediata vecinătate a acesteia pentru identificarea rapidă şi precisă a zonei periculoase.
- ✓ SOFTWARE – Program de funcționare a unui sistem informativ.
- ✓ STARE DE ALARMĂ – starea imediat următoare detectării (înregistrării) apariției unui pericol.
- ✓ TRADUCTOR DE PERICOL – Detector, traductor analogic sau traductor inteligent.
- ✓ UNITATE DE AVERTIZARE ALARMA - Dispozitiv electro-mecanic care converteşte energia in semnale audio-vizuale pentru avertizarea in caz de alarma.
- ✓ WATCH DOG – Detector de timp limită.
- ✓ ZGOMOT – Perturbație aleatoare care apare la ieşirea traductoarelor.
- ✓ ZONĂ DE SUPRAVEGHERE - Spațiul delimitat constructiv şi care poate fi localizat precis în cadrul sistemului de conducere în caz de pericol.

5.2. Centrale de conducere în caz de pericol

Centrala de conducere recepționează semnalele de intrare de la traductoare. Aceste semnale sunt condiționate şi prelucrate pentru determinarea stării de pericol. În funcție de rezultatele obținute se elaborează diferite stări. Pentru acelaşi tip de supraveghere stările de alarmă corespunzătoare au aceiaşi prioritate. În caz contrar starea de alarmă corespunzătoare protecției vieții are prioritate maximă. Semnalele transmise de butoanele acționate manual au prioritate la prelucrare față de semnalele traductoarelor. Această atitudine reflectă concepția că detecția umană a pericolului este superioară celei automate. Starea de alarmă este transmisă sub forma unor semnale de ieşire pe trei căi diferite pentru:

- ♦ Dispozitivele de alarmare locală;

- Dispozitivele de recepție a alarmelor;
- Dispozitivele de protecție automată.

În afară de stările de alarmă există şi stările de defect ale sistemului. Aceste stări sunt transmise la distanță pe o cale specială. Stările de alarmă au prioritate față de stările de defect iar transmisia semnalului de alarmă nu trebuie să fie blocată de starea de defect.

Traductoarele sunt situate, în general, la distanță față de centrală. Conexiunea între ele se face prin linii de legătură. Acestea pot fi electrice, radio sau optice. Pentru liniile electrice defectele sunt următoarele:

- Linie întreruptă;
- Linie în scurtcircuit;
- Linie conectată la pământ.

5.3. Traductoare pentru sisteme de securitate la efracție

Cele mai folosite traductoare pentru sistemele de securitate sunt detectoarele pasive în infraroşu (PIR), monitoarele video de mişcare şi controlerele de acces. Sunt folosite în continuare traductoarele clasice de tip contact (buton de panică, senzor de poziție, etc.) şi a celor de vibrații sau şoc (detector de şoc, detector de geam spart). Acestea sunt foarte utile în special pentru semnalizarea alarmelor preventive.

5.3.1. Detectoare pasive în infraroşu - PIR

Orice corp uman emană o căldură în mediul înconjurător în spectrul infraroşu. Detectorul PIR este un senzor de proximitate bazat pe sesizarea variației termice captate în spectrul infraroşu de către un senzor piroelectric. Alarmele produse de către aceste detectoare nu au un caracter preventiv deoarece ele constată prezența în spațiul protejat şi nu încercarea de pătrundere în acesta. Aceiaşi observație este valabilă şi pentru monitoarele video.

Senzorul piroelectric este format dintr-un fotoelement şi un tranzistor TEC – MOS. Radiația în infraroşu determină fotoelementul să încarce electric capacitatea porții tranzistorului. Acesta va conduce un curent dependent de tensiunea aplicată pe poartă. Pentru a nu fi influențat de variațiile ambiante de radiație în infraroşu senzorul are o construcție diferențială conectată în opoziție. În acest mod valorile absolute ale iluminării ambiante sunt rejectate.

5.3.2. Monitoare video de mişcare

Aceste mai sun cunoscute şi sub denumirea de sisteme de televiziune cu circuit închis şi sunt singurele detectoare de efracție care furnizează la ieşire un semnal analogic în banda video stației centrale de urmărire. Imaginea este captată de către o matrice de fotodiode. Detectoarele video de mişcare sunt comparatoare între două imagini succesive ale senzorului cu fotodiode. Pentru efectuarea comparării una dintre imagini trebuie să fie memorată. Pornirea automată a sistemului de vizualizare şi/sau înregistrare video este dată de ale detectoare de mişcare volumetrice realizate cu ultrasunete, microunde sau PIR. Scanere video realizează compararea imaginii captată de către senzor cu cele existente într-o bancă de date.

Datorită volumului mare de prelucrări ale imaginilor dinamice monitoarele video de mişcare sunt echipamente lente.

5.3.3. Controlere de acces

Controlerele de acces sunt echipamente electronice ataşate unor căi fizice de acces pentru a permite intrarea şi/sau ieşirea autorizată pentru zona protejată, Fig. 5.1. Căile fizice de acces pot fi porți, uşi, lifturi automate de livrare a unui serviciu, etc. Autorizarea se poate face prin mai multe metode care diferențiază sistemele între ele. Una dintre metode foloseşte un cod de acces introdus cu ajutorul unei tastaturi. Un cod corect determină deblocarea accesului pentru un timp limitat.

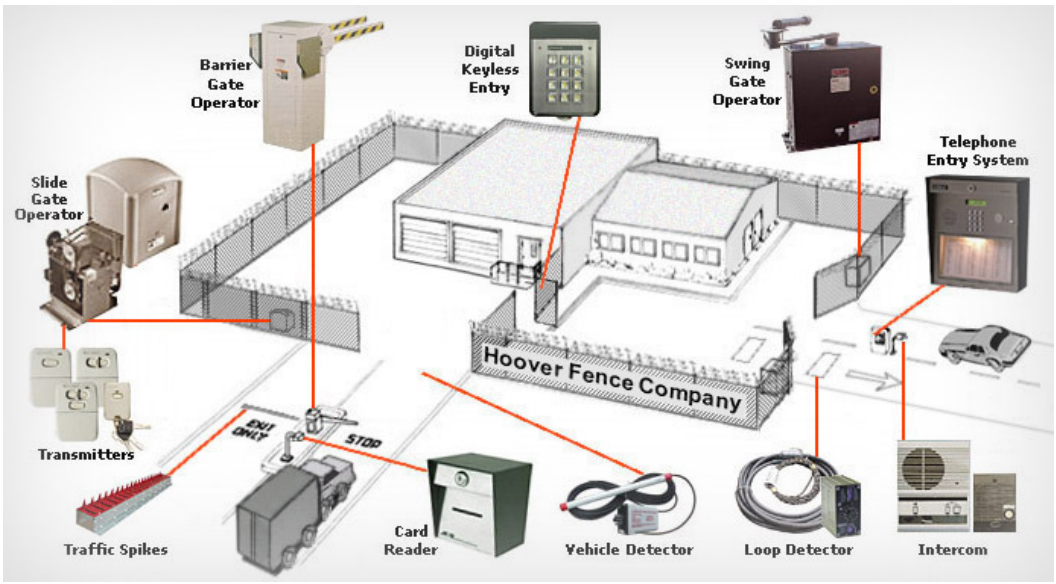


Fig. 5.1 Un sistem de control al accesului

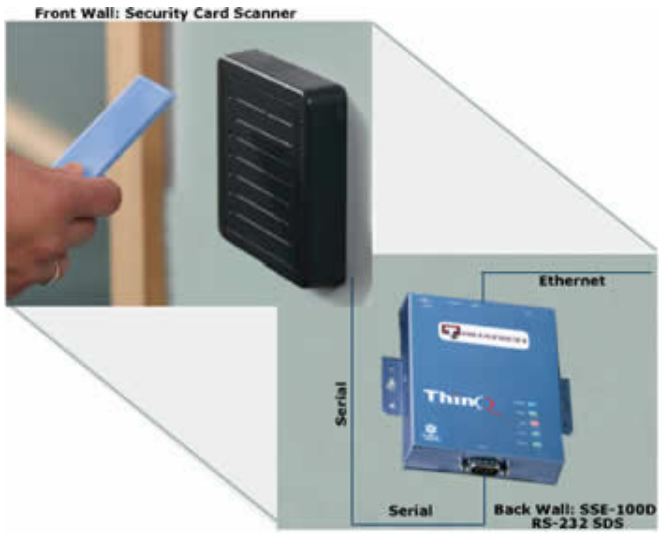


Fig. 5.2 Un cititor de cartele pentru controlul accesului încadrat într-un sistem de domotică

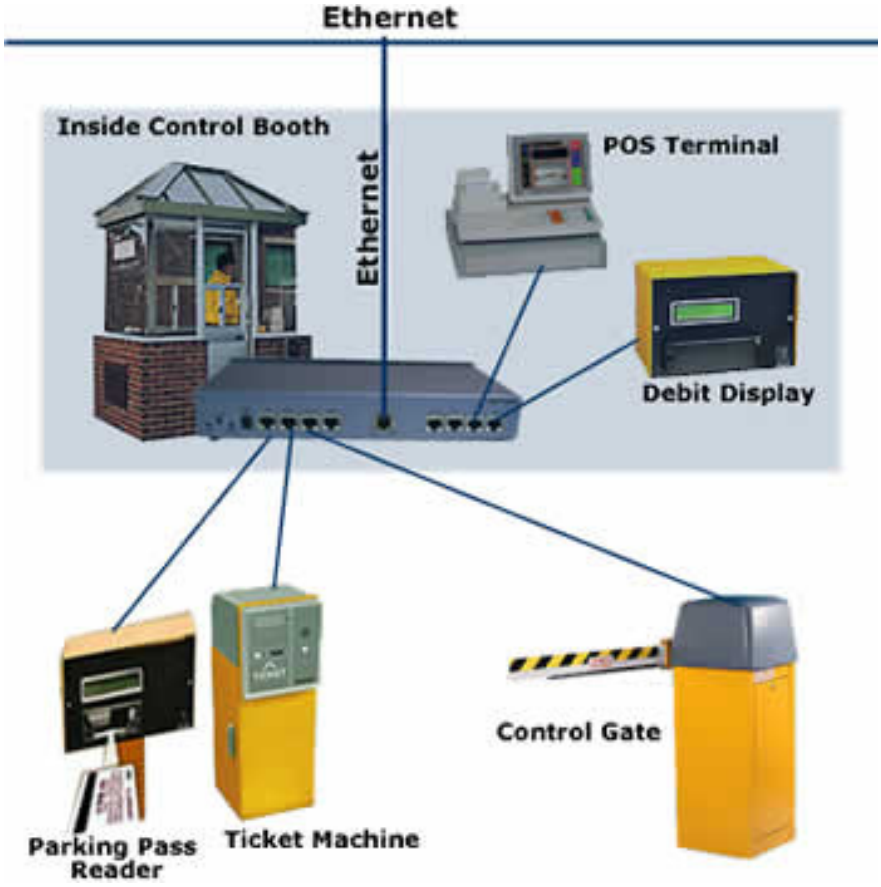


Fig. 5.3 O barieră pentru controlul accesului vehicolelor încadrat într-un sistem de domotică

Toate sistemele de control a accesului prezentate beneficiază de aportul unei baze de date referitoare la codurile persoanelor şi vehicolelor. După examinarea bazei de date se pot lua diferite acțiuni, de exemplu blocarea unor bariere. Din acest punct de vedere sunt niste sisteme de gestionare a clădirilo, BMS. Se observă, pe de altă parte ca aceste sisteme au o structură ierarhică şi din această cauză sunt şi sisteme de domotică.

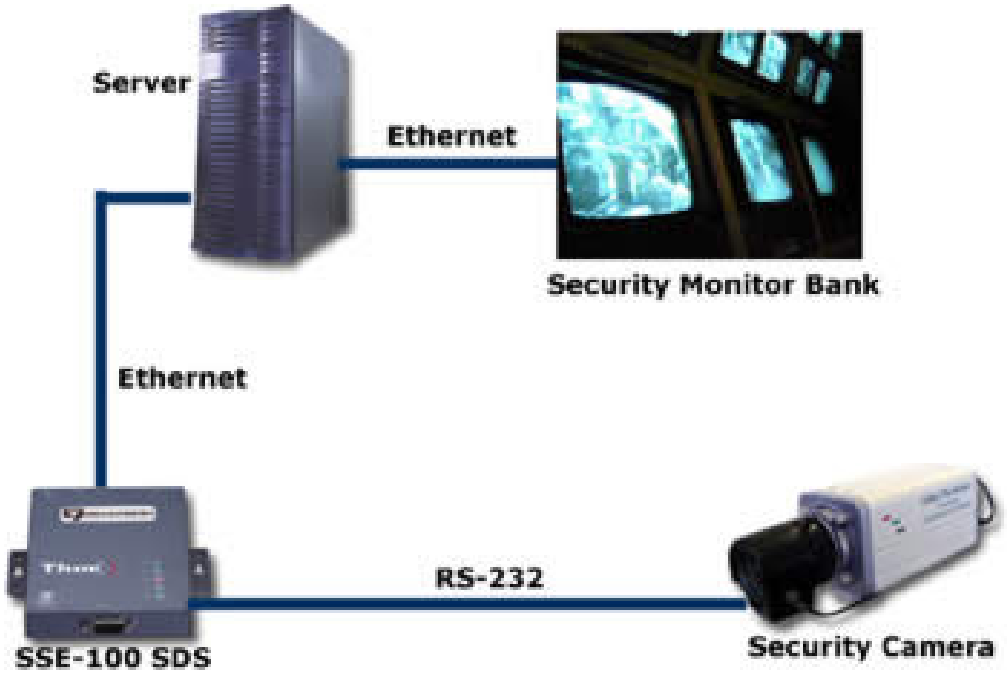


Fig. 5.4 Un sistem domotic de securitate cu cameră video

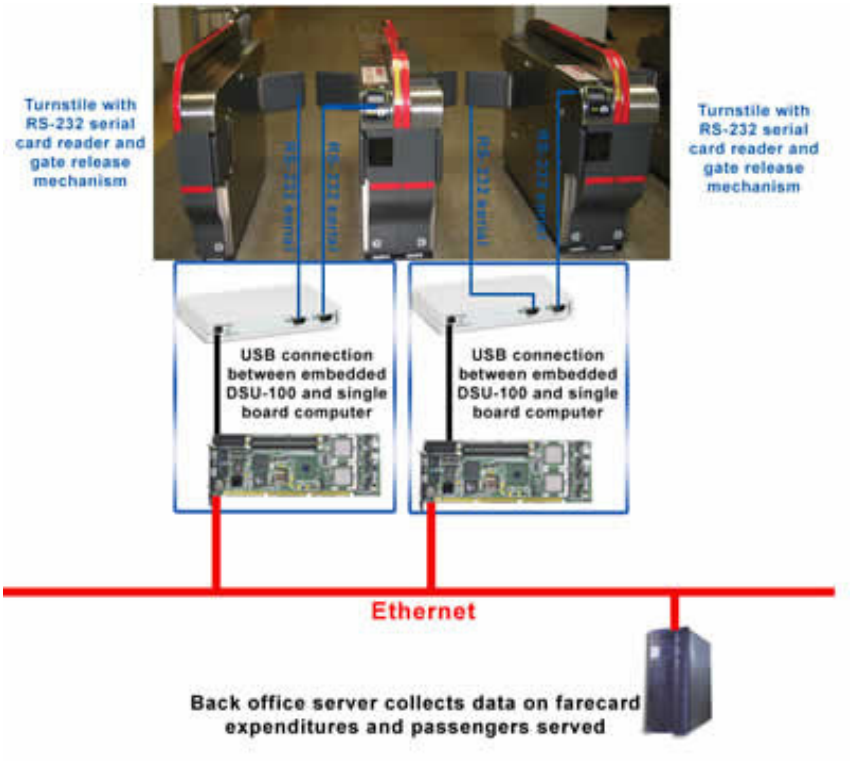

Turnstile with RS-232 serial card reader and gate release mechanism
Turnstile with RS-232 serial card reader and gate release mechanism
USB connection between embedded DSU-100 and single board computer
USB connection between embedded DSU-100 and single board computer
Ethernet
Back office server collects data on farecard expenditures and passengers served

Fig. 5.5 O barieră pentru controlul accesului persoanelor încadrată într-un sistem de domotică

5.3.4. Sisteme de identificare cu frecvență radio

Una din ultimele apariții in lumea tehnologica actuala este RFID (Radio Frecquency IDentification) sau identificarea de proximitate.

RFID este o tehnologie avansata de colectare automata a datelor, aceasta tehnologie seamănă foarte mult cu sistemele bazate pe coduri de bare deoarece in mare au aceleaşi componente deşi au principii de funcționare diferite .

Tehnologia codurilor de bare presupune un cititor optic (bazat pe raze laser) si o eticheta ce prezintă o succesiune de linii ataşata unui obiect. Spre deosebire de aceasta, RFID foloseşte un cititor (bazat pe un emițător de frecvente radio in banda joasa) si un microcip (transponder sau tag) care poate fi implementat in interiorul obiectului sau pe o cartela ataşate de acesta.

Cititorul emite un câmp electromagnetic pe care tranponderul îl preia si prelucrându-l îl retrimite înapoi cititorului sub forma unor impulsuri, comunicarea intre tag si cititor făcându-se in fracțiuni de secunda, Fig. 5.6.

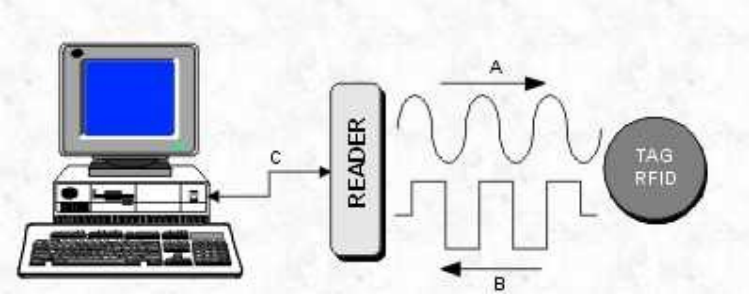


Fig. 5.6 Sistem cu traductor de proximitate

Un sistem RFID este compus după cum se poate observa in figura prezentata mai sus dintr-un cititor, un calculator si transponderul respective.

Datorita faptului ca nu este nevoie a se efectua un contact direct intre transponder si cititor acestea pot lucra in condiții foarte vitrege (medii umede, uleioase, medii cu mult praf si mizerie) fiind foarte rezistente la temperaturi joase (40°C) si înalte (+200°C).

Tag-ul si cititorul pot fi separate de materiale textile sau medii nemetalice, transmiterea impulsurilor dintre ele efectuându-se prin acestea.

Cele mai simple aplicații ale detectării de proximitate se pot compara cu sistemele bazate pe coduri de bare, insa profitabilitatea acestui sistem se observa in aplicații precum GPS (Global Positioning Satelite system).

Alte aplicații pentru acesta tehnologie se bazează pe plasarea tag-ului in interiorul pneurilor auto, obținându-se astfel informații rapide despre producător, data fabricației, lotul de produse, locul comercializări etc sau implementarea tag-ului in cartele folosite la sistemele de acces in instituții si restricționarea accesului pe nivele de competenta.

5.4. Traductoare pentru sisteme de siguranță la foc

Detectoarele automate de incendiu sunt elemente periferice ale instalațiilor de semnalizare a incendiilor prin care se supraveghează in mod continuu sau la anumite intervale de timp un parametru fizic şi/sau chimic asociat incendiului.

In caz de incendiu, detectoarele declanşează un semnal care este transmis la centrala prin intermediul circuitelor de legătura.

Oricare ar fi tipul de detector, rolul sau intr-o instalație de semnalizare consta in a depista si semnaliza cat mai repede incendiul.

Pentru a acționa eficient, un detector automat de incendiu trebuie sa îndeplinească, in principal, următoarele caracteristici:

- ✓ funcționare sigura in condiții specifice de mediu (temperatura, umiditate, curenti de aer, concentrații de praf etc.)
- ✓ timp de răspuns rapid in prezenta parametrului supravegheat;
- ✓ stabilitate in timp a pragului de acționare;
- ✓ temporizare pentru eliminarea semnalizărilor false;
- ✓ imunitate la semnale perturbatoare;
- ✓ consum propriu redus de energie;
- ✓ semnalizarea stării0de buna funcționare (veghe);
- ✓ construcție simpla;
- ✓ întreținere si depanare uşoara.

Pentru a indica intrarea in stare de alarma, detectoarele de incendiu trebuie sa fie prevăzute cu semnalizare optica locala. Dispozitivele optice utilizate in acest scop trebuie sa emită lumina de culoare roşie, uşor vizibila de la distanta. Unele tipuri de detectoare mai au prevăzut suplimenta, fata de semnalizarea optica locala, un circuit separat de semnalizare la distanta. Acest circuit este necesar pentru punerea in funcțiune a unei lămpi de semnalizare montata intr-un loc uşor vizibil, pentru situațiile in care detectoarele sunt instalate in locuri greu accesibile.

In construcția detectoarelor, trebuie avut in vedere ca eventualele defecțiuni ale circuitelor, care pot scoate din funcțiune aparatul sau împiedica inițierea si transmiterea semnalului de incendiu la centrala, sa fie semnalizate local şi/sau la centrala, ca stare, de defect.

In general, utilizarea dispozitivelor electromecanice trebuie evitata. In cazuri speciale, când acest lucru este necesar, pentru transmiterea semnalului de incendiu, trebuie adoptata varianta cu contact normal închis in stare de veghe.

Soclurile necesare instalării detectoarelor trebuie sa aibă o singura poziție de fixare si sa fie cu contacte autocuratitoare. Sistemul de prindere a conductelor aferente circuitelor electrice trebuie sa asigure un contact electric sigur.

Detectoarele de incendiu se pot clasifica, in principal, după următoarele criterii:

- ✓ In funcție de parametrul supravegheat:

a). detector de temperatura: sensibil la temperatura şi/sau gradient de temperatura şi/sau diferența de temperatura;

b). detector de fum: sensibil la particulele produse de combustie şi/sau piroliza, suspendate in atmosfera:

detector cu camere de ionizare: sensibil la particulele capabile sa afecteze curentul de ionizare

detector optic: sensibil la particulele capabile sa afecteze absorbția sau împrăştierea radiațiilor din spectrul infraroşu şi/sau vizibil şi/sau ultraviolet;

c). detector de gaze de combustie: sensibil la anumite produse gazoase rezultate in urma combustiei şi/sau descompunerii termice;

d). detector de flacăra: sensibil la radiația electromagnetica emisa de flăcările de incendiu.

- ✓ In funcție de modul de răspuns la parametrul supravegheat:

  a). detector cu acționare statica: semnalizează la atingerea unei valori prestabilite a parametrului supravegheat;

  b). detectoare cu acționare diferențiala: semnalizează la depăşirea unei valori prestabilite a diferenței de mărime a parametrului supravegheat in cel puțin doua locuri;

  c). detector cu acțiune velocimetrică: semnalizează la depăşirea unei valori prestabilite a vitezei de creştere (gradient) a parametrului supravegheat.

- ✓ In funcție de configurația senzorului:

  a). detector punctual: acționează la parametru supravegheat din vecinătatea unui senzor punctual;

  b). detector multipunctual: acționează la parametrul supravegheat din vecinătatea mai multor puncte;

  c). detector liniar: acționează la parametrul supravegheat din vecinătatea unei linii continue.

- ✓ In funcție de modul de reutilizare după acționare (producere alarma, de verificare):

  a). detector reutilizabil: poate fi readus in stare de funcționare, in vederea unei noi acționari, după încetarea condițiilor care au produs acționarea sa, fără înlocuirea vreunei componente;

  b). detector parțial reutilizabil: poate fi readus in stare de funcționare, după încetarea condițiilor care au produs acționarea sa, prin înlocuirea unor componente;

  c). detector nereutilizabil: care nu mai poate fi reutilizat după acționare, fiind necesara înlocuirea sa.

Clasificările de mai sus nu sunt limitative, putând exista si diverse combinații ale tipurilor prezentate sau funcționând pe alte principii sau alte criterii de clasificare.

5.4.1. Butoane manuale de semnalizare.

Butonul manual de semnalizare reprezintă dispozitivul prin intermediul căruia se poate semnaliza manual, de către om, apariția unui incendiu. Cu toata răspândirea din ce in ce mai mare a detectoarelor automate de incendiu in instalațiile de semnalizare, butoanele manuale de semnalizare sunt folosite, încă, pe scara larga, deoarece prezintă o construcție simpla si siguranța ridicata in exploatare. Folosirea butoanelor manuale in cadrul instalațiilor automate de semnalizare a incendiilor este justificata si prin faptul ca, in anumite situații, incendiul poate fi observat de către om înainte de declanşarea unui detector automat si ca atare este rațional ca instalațiile sa se prevadă si cu aceasta posibilitate.

Instalațiile de semnalizare a incendiului se prevăd numai cu acționare manuala doar in acele situații in care intervenția pentru stingerea in caz de incendiu se asigura in timp util.

Butoanele de semnalizare a incendiilor se vor amplasa in locuri vizibile, usor accesibile, de preferința lângă uşa, la intrarea in casa scărilor sau in aceasta si in general in punctual de circulație obligatorie in caz de evacuare. In cazul spatiilor cu suprafețe mari de supraveghere (încăperi, culoare, hale de producție etc.), butoanele de semnalizare se vor amplasa astfel încât nici o persoana sa nu aibă nevoie a se deplasa mai mult de circa 50 m, de la orice poziție din clădire, spre a da alarma de incendiu. Butoanele de semnalizare se amplasează de regula la o înălțime de circa 1,4 m de la pardoseala. Atunci când este necesar, locul de amplasare al acestora va fi iluminat corespunzător pentru a fi uşor observata. Pentru clădirile cu mai multe nivele, butoanele de semnalizare se vor amplasa la fiecare nivel, in apropierea scărilor sau a altor cai de acces. Nu este admisa conectarea butoanelor de semnalizare, de pe diferite nivele, la acelaşi circuit de linie din centrala de semnalizare.

Spatiile in care se prevăd detectoare automate, conform legislației in vigoare, vor fi dotate in mod obligatoriu si cu butoane manuale de semnalizare, instalate pe circuite de linii distincte. La baza acestei prevederi, a stat considerentul realizării unei siguranțe ridicate in semnalizarea apariției unui incendiu prin semnalizare manuala, de către om, înainte de acționarea unui detector automat de incendiu, pe de o parte, cat si existenta unei rezervări in cazul in care circuitul de linie cu detector automat ar fi defect, pe de alta parte.

Principiul care sta la baza funcționarii butoanelor de semnalizare manuala este mecanic si consta, in funcție de varianta constructiva a aparatului, in închiderea sau deschiderea unor contacte. Datorita siguranței mai ridicate in transmiterea semnalizării de incendiu, in instalațiile de semnalizare, se utilizează cu precădere butoanele de semnalizare care – in starea normala de veghe – prezintă un contact normal închis (CNI) si in alarma – contact normal deschis (CND). Aceasta cerința a rezultat din practica, unde s-a constatat ca – datorita unei întrețineri defectuoase in special a elementelor de etanşeizare in locurile cu mult praf, umezeala, substanțe corozive etc. – transmiterea semnalizării de incendiu nu s-a mai putut face la acționarea butonului, datorita oxidării sau depunerii prafului pe contactele din interiorul aparatului.

La unele tipuri de butoane de semnalizare, exista si posibilitatea realizării unei legaturi fonice cu centrala de semnalizare, legătura ce se stabileşte in mod automat după acționarea acestuia pentru transmiterea semnalizării de incendiu.

Aceste tipuri sunt deosebit de utile, întrucât operatorul de serviciu se poate informa cu date privind natura si amploarea incendiului.

Din punct de vedere al construcției, butoanele de semnalizare se fabrica in variantele:

pentru medii normale, destinate amplasării in interiorul construcțiilor sau in exteriorul acestora;

pentru medii explozive (de interior si de exterior);

pentru mediu naval (de interior si exterior).

5.4.2. Detectoare de temperatura

Cele mai răspândite detectoare de incendiu aflate in exploatare in instalațiile de semnalizare a incendiilor sunt detectoarele termice. Acest fapt se datorează in primul rind simplității, robusteții si prețului relativ scăzut al aparatelor. Cu toate acestea, detectoarele care funcționează cu fir sau aliaj fuzibil au si anumite inconveniente care nu pot fi neglijate.

Dintre acestea, cele mai importante sunt:

- inerția termica a aliajului nu asigura sesizarea destul de rapida a apariției incendiului, in special daca gazele fierbinți nu se propaga direct spre detector;
- aria de supraveghere este redusa, ceea ce face sa fie necesara montarea unui numar mare de detectoare;
- temperatura mediului ambiant influențează timpul de răspuns.

Detectoarele de temperatura cu elemente de bimetal cunosc, de asemenea, o mare răspândire. Funcționarea acestora se bazează pe proprietatea dilatării inegale a doua metale cu coeficienți diferiți de dilatare (de exemplu: alama) sudate pe suprafața. Prin încălzirea lamei, datorita dilatării inegale a celor doua fete, aceasta se va curba in direcția lamei cu coeficient de dilatare mai mic, si, in funcție de tipul constructiv, se va închide sau deschide un contact electric.

Spre deosebire de detectoarele cu fuzibil, la cele cu bimetal temperatura de declanşare poate fi reglata in limite foarte largi.

Pentru a elimina dezavantajele legate de influenta temperaturii inițiale a mediului asupra detectorului, se utilizează detectoarele diferențiate si velocimetrice. Indiferent de principiul de funcționare, detectorul termodiferential inițiază un semnal de alarma atunci când diferențele, normal mici, de temperatura, in doua sau mai multe locuri, depăşesc – pentru o durata de timp suficienta – o anumita valoare prestabilita.

Detectoarele termovelocimetrice funcționează pe principiul măsurării vitezei de creştere a temperaturii in unitatea de timp.

Sensibilitatea acestor aparate este de ordinul a 5-8 °C/minut. Detectoarele termodiferentiale si termovelocimetrice sunt mai sensibile si funcționează mai rapid decât cele termostatice (de maxim). In cele mai multe situații, aceste tipuri de detectoare sunt combinate si cu funcțiunea de semnalizare termostatică (de maxim).

Detectoarele pneumatice de temperatura se compun, in principiu, din doua camere de aer suprapuse. Una din camere (compensare) este izolata fata de atmosfera înconjurătoare. Cealaltă camera (receptoare0 este in legătura directa cu atmosfera. Intre cele doua camere, se afla o membrana metalica elastica care formează primul electrod. In stare normala, membrana se afla in contact cu un vârf metalic prevăzut cu un şurub de reglaj, care formează cel de-al doilea electrod.

Când temperatura mediului creste încet, presiunea aerului in cele doua camere se echilibrează prin intermediul ajutajului care face legătura intre acestea.

In caz de incendiu, temperatura creste brusc si – implicit – membrana este deplasata înspre camera de compensare, întrerupând contactul cu cel de-al doilea electrod.

Si aceste tipuri de detectoare, in cele mai multe cazuri, sunt asociate cu funcţia termostatică.

Detectoarele de temperatura cu aliaj fuzibil, bimetal si pneumatic fac parte din categoria detectoarelor care nu consuma energie electrica in funcţionare.

Pentru a elimina dezavantajele legate de inertia termica relativ mare a senzorilor prezentaţi anterior, s-au construit detectoare cu elemente semiconductoare. Dintre acestea, cel mai uzual sunt folosite termorezistenţele si termistorii. De regula, aceste detectoare îndeplinesc o funcţie dubla, termostatică si termovelocimetrica.

Funcţionarea acestor tipuri de detectoare se bazează pe variaţia rezistentei senzorilor sub influenta temperaturii. Variaţia de rezistenta este prelucrata de circuite electronice si in final transmisa sub forma de semnal de alarma la centrala de semnalizare.

In fig. 3 se prezintă structura unui detector de temperatura cu polimeri semiconductori, denumit Alarm Line Detector.

Patru conductori metalici, fiecare acoperit cu o izolaţie de material plastic semiconductor, sunt plasaţi intr-o manta protectoare de PVC rezistent la temperatura ridicata.

In general, lungimea standard a cablului care se livrează este de 200 m, diametrul exterior al cablului fiind de 3,0 … 4,5 mm. Conductoarele sunt astfel legate încât se formează doua circuite electrice distincte. Prin intermediul centralei de semnalizare, special construita pentru acest tip de detector, se supraveghează integritatea la o eventuala rupere a unuia dintre cei 4 conductori. Creşterea temperaturii peste o anumita limita conduce la scăderea rezistentei de izolaţie dintre cele doua circuite electrice distincte. Rezistenta de izolaţie este continuu supravegheata de către centrala de semnalizare. La reducerea valorii rezistentei de izolatie sub o anumita valoare, se iniţiază semnalul de incendiu

Acest tip de detector poate fi “echivalat” cu un mare număr de termistori, cu coeficient negativ de temperatura, conectaţi in paralel din punct de vedere electric. Modificarea rezistentei unui grup de termistori, echivalent cu încălzirea unei anumite lungimi de cablu, conduce la scăderea rezistentei echivalente totale care este prelucrata electronic de centrala.

5.4.3. Detectoare de fum cu ionizare

O camera de ionizare tipica consta in doua placi încărcate electric si o sursa radioactiva (in mod obişnuit Americiu 241) pentru ionizarea aerului dintre placi, Fig. 5.7. Sursa radioactiva emite particule care intra in coliziune cu moleculele de aer dislocându-le electronii. O parte din moleculele de aer devin ioni pozitivi prin pierderea electronilor. O alta parte a moleculelor primeşte electroni devenind astfel ioni negativi. Se creează astfel un număr egal de ioni pozitivi si negativi. Ionii pozitivi sunt atraşi de placa negativa, iar cei negativi de placa pozitiva, Fig. 5.8. Acest fenomen generează un curent de ionizare, care poate fi măsurat prin circuitul electronic conectat la cele doua placi.

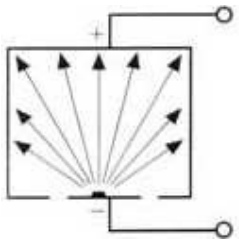


Fig. 5.7 Modelul de radiaţie a particulelor

Particulele rezultate din combustie sunt mult mai mari decât moleculele de aer ionizate. La intrarea particulelor de combustie in camera de ionizare, acestea vor intra in coliziune cu moleculele de aer ionizate si se vor combina cu acestea, Fig. 5.9.

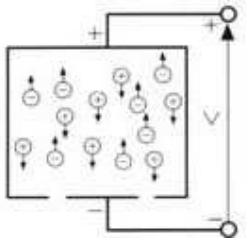


Fig. 5.8 Distribuţia ionilor

Unele particule astfel rezultate vor fi pozitive, iar altele vor fi negative. Aceste particule relativ mari continua sa se combine cu alţi ioni, devin centre de recombinare si totalul particulelor ionizate din cameră se reduce. Reducerea numărului de particule ionizate conduce la scăderea curentului de ionizare mai sus menţionat, care este permanent monitorizat. La scăderea acestui curent sub o valoare predeterminata, se depăşeşte un prag dincolo de care este stabilita condiţia de alarma.

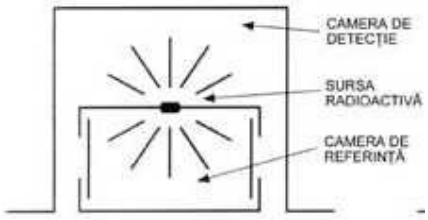


*Fig. 5.9 Camera dual*ă

Schimbările in presiunea si umiditatea atmosferica pot afecta curentul de ionizare si pot crea un efect similar aceluia de pătrundere a particulelor de combustie. Pentru compensarea efectelor nedorite generate de schimbările de presiune si umiditate a apărut camera de ionizare dual*ă*, care se folose*ş*te pe scara larga pe pia*ț*a detectoarelor de fum.

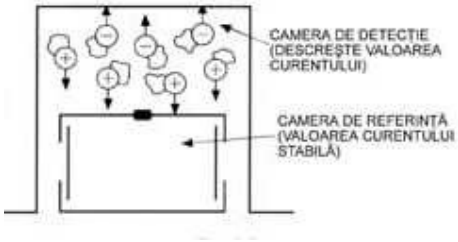


*Fig. 5.10 Cameră dual*ă *cu fum*

Un detector cu camera dual*ă* folose*ş*te doua camere de ionizare. Una dintre ele este camera de detec*ț*ie care este deschisa către aerul din exterior, Fig. 5.9. Camera de detec*ț*ie este afectata de macroparticule, de presiunea si umiditatea atmosferica. Cea de-a doua camera serve*ş*te ca referin*ț*a si este par*ț*ial închisa fata de aerul din exterior. Ea este afectata numai de presiunea si umiditatea atmosferica întrucât deschiderile mici prin care pătrunde aerul nu permit intrarea macroparticulelor cum ar fi particulele de fum. Circuitul electronic monitorizează ambele camere si compara semnalele, Fig. 5.10.

Fig. 5.11 Detector cu obturarea luminii

Schimbările de presiune sau umiditate ale atmosferei afectează in mod identic semnalele de ie*ş*ire ale ambelor camere de ionizare, care in acest fel se anulează reciproc. La intrarea particulelor de fum in camera de detec*ț*ie scade curentul de ionizare al acesteia comparativ cu curentul din camera de referin*ț*a, care virtual rămâne neschimbat. Diferen*ț*a de curent rezultata este detectata de circuitul electronic.

Exista totuşi o serie de probleme care pot afecta detectoarele cu camera dual㠖 de ionizare: praful, umiditatea excesiva (condensul), curenţi de aer semnificativi, mici insecte care pot fi "citite" ca particule de combustie de către circuitul electronic al detectorului. Cu cat detectorul este calibrat mai sensibil cu atât mai mult aceste probleme pot afecta performantele detectorului generând alarme false.

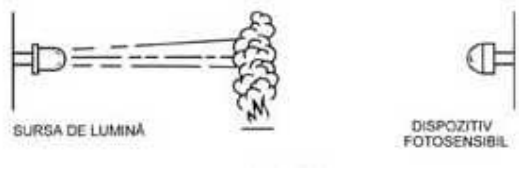


Fig. 5.12 Detector cu obturarea luminii în prezenţa fumului

5.4.4. Detectoare de fum optice

La aceste detectoare fumul afectează intensitatea unui fascicol de lumina ce trece prin aer. Fumul poate obtura sau chiar bloca acest fascicol. De asemenea poate cauza difuzia luminii datorita reflexiilor pe particulele de fum. Detectoarele de fum fotoelectrice sunt proiectate deci pentru a sesiza fumul folosind aceste efecte ale fumului asupra luminii.

5.4.5. Detectoare de fum fotoelectrice cu obturarea luminii

Unul dintre tipurile de baza de detector de fum fotoelectric este detectorul cu obturarea luminii. Acesta este format dintr-o sursa de lumina si un dispozitiv fotosensibil, cum ar fi o fotodioda, Fig. 5.12. Semnalul de ieşire al elementului fotosensibil este afectat de particulele de fum care blochează parţial fascicolul, Fig. 5.13. Schimbarea acestui semnal este sesizata de circuitul electronic al detectorului si daca se depăşeşte un anumit prag se generează semnal de alarma. Detectoarele cu obturare sunt de obicei de tip cu fascicul proiectat, la care sursa de lumina extinde aria de protejat.

5.4.6. Detectorare de fum fotoelectrice cu difuzarea luminii

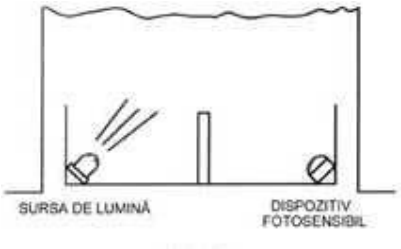


Fig. 5.13 Detector cu difuzia luminii

Majoritatea detectoarelor de fum sunt de tip focalizat si operează pe principiul luminii difuzate. O dioda LED luminează o arie care uzual nu este "văzuta" de elementul fotosensibil, care de obicei este o fotodioda, Fig. 5.13. La pătrunderea particulelor in calea luminii, aceasta este reflectata pe elementul fotosensibil, Fig. 5.14, activând detectorul.

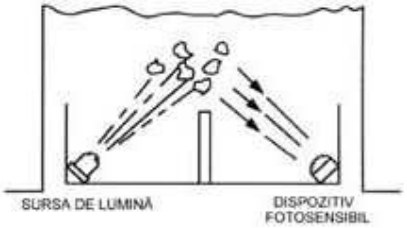


Fig. 5.14 Detector cu difuzia luminii în prezența fumului

5.4.7. Amplasarea detectoarelor de pericol la incendiu

Cu cat este mai repede identificata sursa unei alarme cu atat mai repede se poate actiona. Desi nu exista reguli formale de zonare in temeni de protectie la foc (cu exceptia dispozitivelor fara fir, car etrebuie sa aiba fiecare adresa proprie), aceasta este o operatie de finete. Instalatorii cu experienta si proiectantii de sisteme recomanda urmatoarele:

- Stabiliti cel putin o zona la fiecare nivel protejat.
- Zonati subdiviziunile naturale ale unei cladiri mari, cum ar fi aripile separate de pe acelasi nivel.
- Reduceti la minim numarul detectoarelor de pe fiecare zona. Mai putine detectoare pe o zona vor rezulta in localizarea mai rapida a focului si simplificarea depanarii.
- Instalati detectoare in conducta in alte zone decat acelea in care sunt instalate detectoare pentru spatiu deschis, in scopul localizarii si depanarii cat mai precise.
- Deseori detectoarele de fum sunt folosite pentru a controla echipamente auxiliare. O parte dintre aplicatiile tipice sunt dupa cum urmeaza:
- Controlul evacuarii fumului in sistemele de ventilatie si de conditionare a aerului.
- Deschiderea usilor pentru evacuarea fumului in situatii de foc.
- Deschiderea usilor pentru a permite iesirea in situatii de foc.
- Oprirea lifturilor in caz de foc.
- Actionarea unui sistem de inchidere.

Majoritatea detectoarelor folosite in instalatiile de stingere au contacte de releu auxiliare, care sunt direct conectate la sistemul sau dispozitivul de controlat. Aceste dispozitive trebuie aprobate in mod special pentru a fi folosite in comanda stingerii.

Mai mult decat atat, pentru detectoarele folosite in instalatiile de stingere spatierea si cerintele de amplasare pot fi diferite de detectoarele folosite in aplicatii conventionale cu spatii deschise. Se recomanda folosirea detectoarelor pe 4 fire in aceste situatii, intrucat, in functie de centrala de control si de detectoarele folosite, un singur detector poate primi suficienta putere pe un circuit de 2 fire, pentru a opera in cazul unei alarme.

Caracteristicile detectoarelor cu ionizare le fac mai potrivite pentru detectia focului cu flacara (uzual provenit din materiale foarte inflamabile) caracterizat de particule de combustie in plaja 0,01 - 0,03 microni.

Detectoarele fotoelectrice, pe de alta parte, sunt mai potrivite pentru detectia focului cu ardere mocnita (uzual provenit din hartie, materiale textile etc.) caracterizat de particule de combustie in plaja 0,3 - 10,00 microni. Totusi, ambele tipuri de detector pot sesiza cele doua tipuri de foc, dar timpul de raspuns va fi variabil, in functie de tipul de foc.

Deoarece cladirile protejate contin o mare variatate de combustibili, este dificil de prevazut care tip de macroparticule vor aparea in eventualitatea unui incendiu. De asemenea, lucrurile se complica si tinind cont de faptul ca pentru acelasi combustibil, diferite surse de aprindere pot avea efecte diferite. De exemplu, o tigara aprinsa aruncata pe o canapea sau pe

un pat va produce, de obicei, un foc cu ardere mocnita. Totusi, daca tigara cade mai intai pe un ziar aruncat pe canapea sau pat, focul rezultat va fi caracterizat mai mult de flacari decat de ardere mocnita.

Pentru a asigura o avertizare incipienta in cazul unui incendiu, detectoarele (de flacara, de fum) vor fi amplasate in toate zonele cu premisa de protectie. Acoperirea totala, conform standardelor americane, va include toate camerele, coridoarele, zonele de depozitare, subsolurile, podurile si spatiile de deasupra tavanelor false, inclusiv zonele folosite ca parte a sistemului de ventilatie si conditionare a aerului. In plus trebuie protejate debaralele, casa liftului, scarile inchise, jgheaburile si alte subdiviziuni si spatii accesibile.

Este posibil ca sistemele de detectie a incendiului instalate pentru a fi in conformitate cu normativele locale sa nu fie adecvate pentru o avertizare rapida. Unele normative pot avea obiective minime, cum ar fi oprirea lifturilor sau prevenirea circulatiei fumului prin sistemul de ventilatie si conditionare a aerului, in locul avertizarii incipiente.

La instalarea unui sistem de detectie a incendiului de orice tip, utilizatorultrebuie sa puna in balanta costurile si beneficiile acestuia. Localizarea, numarul detectoarelor si zonarea acestora vor fi determinate mai degraba de obiectivele dorite decat de cerintele minime sau de normativele locale .

Se poate renunta la detectie in spatiile inflamabile oarbe, unde se indeplineste una dintre urmatoarele conditii:

- Unde tavanul unui spatiu ascuns este atasat direct la partea de dedesubt a grinzilor de sustinere a unui acoperis sau suport de acoperis, inflamabil.
- Unde spatiul ascuns este complet umplut cu izolatie neinflamabila (in constructii din profile laminate izolatia trebuie sa umple numai spatiul dintre tavan si marginea de jos a profilului acoperitilui sau suportului acoperisului).
- Unde spatiile ascunse de deasupra camerelor nu depasesc 4,6 m2.
- In spatiile formate de ansambluri de stalpi de sustinere aparenti sau profile solide, in pereti, dusumele sau tavane, unde distanta intre stalpi nu depaseste 15 cm.

De asemenea, se poate renunta la detectoare sub tavanele deschise tip plasa, unde sunt indeplinite urmatoarele conditii:

- Deschiderile plasei sunt cel putin 6 mm. dimensiune minima.
- Grosimea materialului nu depaseste cele mai mici ochiuri ale plasei.
- Ochiurile constituie cel putin 70% din suprafata materialului din care este facut tavanul.

In mod uzual detectia este recomandata dedesubtul docurilor sau platformelor de incarcare, dedesubtul acoperisurilor acestora. De asemenea se recomanda instalarea detectoarelor in zonele de sub dusumele in cladirile fara subsoluri.

Totusi, se poate renunta la detectie in spatiile inflamabile oarbe, unde se indeplineste una dintre urmatoarele conditii:

- Spatiul nu se preteaza depozitarii, este protejat la intrarea persoanelor neautorizate si contra acumularii reziduurilor purtate de vant.
- Spatiul nu contine echipament sau structuri potential inflamabile (cum ar fi conducte de abur, cabluri electrice, conducte, coloane sau benzi transportoare) sau care pot ajuta la raspandirea focului.
- Dusumeaua de deasupra spatiului respectiv este foarte aproape.
- In spatiul de deasupra dusumelei respective se manevreaza, se prelucreaza sau se depoziteaza lichide neinflamabile.

Un sistem complet de detectie a incendiului este ceea ce se numeste "cu acoperire totala". In anumite zone, cum ar fi poduri, debarale, dedesubtul docurilor sau platformelor de incarcare, un detector de temperatura poate fi mai potrivit decat un detector de fum. Se va acorda mare atentie instructiunilor fabricantului si urmatoarelor recomandari din acest ghid.

In general, intr-o incapere unde este necesar un singur detector, acesta trebuie plasat cat mai aproape posibil de centrul tavanului. Aceasta localizare este optima pentru detectarea fumului din orice parte a incaperii. Daca nu este posibila o localizare centrala, detectorul poate fi amplasat la o distanta de cel mult 10cm. de perete sau, daca este un detector cu montaj pe perete, poate fi montat chiar pe perete. Pentru detectoarele care se monteaza pe perete localizarea se face asa incat intre partea superioara a detectorului si tavan sa fie circa 10 - 30cm., iar distanta pana la cel mai apropiat colt sa fie de maxim 10cm. (vezi Fig. 3.18).

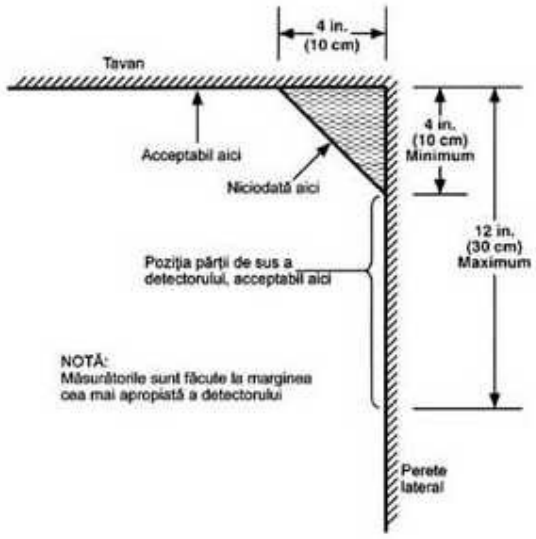


Fig. 5.15 Detector de incendiu montat pe perete

Când intr-o încăpere se afla pe tavan conductele de tur/retur aer detectorul trebuie plasat in calea curgerii aerului către conducta de retur, Fig. 5.15.

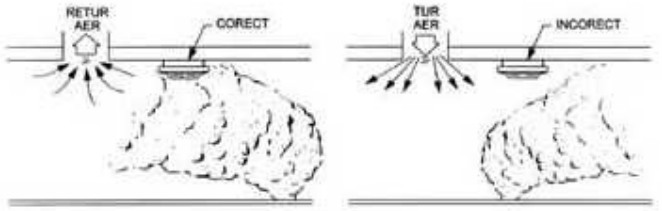


Fig. 5.16 Amplasarea detectoarelor de incendiu lângă conductele de intrare / ieşire aer

Testele cu fum sunt de mare ajutor in determinarea poziției corespunzătoare a detectorului. Trebuie luata in calcul direcția si viteza fumului întrucât acestea pot afecta performantele detectorului. Plasarea detectoarelor lângă aerul condiționat sau lângă gurile de ventilație poate cauza acumulări excesive de praf si murdărie. Acesta acumulare poate avea ca efect funcționarea defectuoasa a detectoarelor generând alarme false. In aceste condiții detectorul nu va fi poziționat la o distanta mai mica de 90cm. de difuzorul de intrare a aerului.

In sistemele proiectate corespunzător, detectoarele focalizate pot fi, de asemenea, localizate in conductele de retur aer sau in carcase tipice de detector in conducta, proiectate special pentru astfel de aplicații. Deşi aceste detectoare particulare nu constituie un înlocuitor al detectoarelor pentru zone deschise, ele asigura o metoda eficienta de acționare a dispozitivelor de control din clădire pentru a preveni transportarea fumului din zona de foc in alte parți ale clădirii.

Una dintre cauzele majore ale alarmelor false este amplasarea necorespunzătoare a detectoarelor. Cea mai buna cale de evitare a alarmelor false este evitarea instalării detectoarelor in medii care pot produce o funcționare defectuoasa. Aceste medii pot fi:

- ✓ Zone cu praf si murdărie excesiva cum ar fi camere de alimentare, oțelarii etc. Praful si murdăria se pot acumula in camera de detecție a senzorilor si-i pot face foarte sensibili, sau, dimpotrivă, pot bloca intrarea aerului in camera de detecție, reducând astfel sensibilitatea senzorilor. Trebuie acordata o mare atenție evitării zonelor unde se folosesc echipamente ce produc abur, sau ceata artificiala, sau componente utilizate la spălat si curățat. Aceste substanțe pot genera alarme false.
- ✓ Zone de exterior, grajduri, şoproane deschis pentru depozitare sau alte structuri deschise, afectate de praf, curenți de aer sau umiditate si temperatura excesive.
- ✓ Zone umede sau cu umiditate excesiva sau in imediata apropiere a camerelor de baie cu dus. Picaturile de apa se pot acumula in camera de detecție a senzorului si acesta poate deveni excesiv de sensibil. In timpul unui dus fierbinte se produce o cantitate imensa de aer umed. Umezeala din acest aer poate intra in camera de

detecție ca vapori de apa, care se răcesc si condensează in mici picaturi ce generează alarme false.

- ✓ Holurile de aşteptare a liftului - nu poziționați detectoare deasupra scrumierelor sau in locurile unde se fumează in aşteptarea liftului.
- ✓ Medii foarte reci sau foarte fierbinți sau in clădiri neîncălzite sau camere unde temperatura poate ieşi din plaja de temperatura a detectorului (fie sub limita minima fie peste limita maxim admisa).

5.5. Sistem simplu de securitate

Un sistem simplu de securitate cu două detectoare t1 şi t2 la intrare şi hupa H la ieşire este prezentat în Fig. 5.17. Pentru a se asigura semnalizarea în cazul întreruperii unei legături de la detectoare (fail-safe) contactele sunt normal închise.

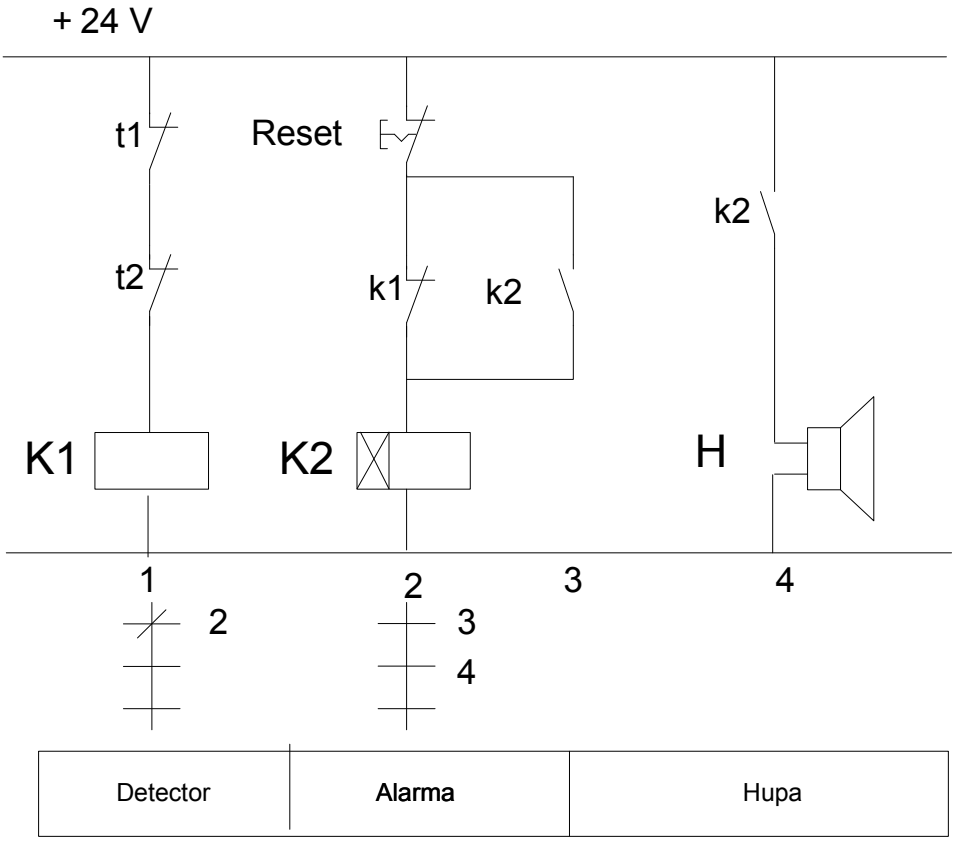


Fig. 5.17 Sistem de securitate fail-safe

5.6. Sistemul MicroSAM

Micro Supervizorul Adresabil Modular MicroSAM, produs în România de AUTOMATICA SA, este un *sistem de tip adresabil* destinat supravegherii, detectării şi alarmării în cazul apariţiei unor pericole în clădiri şi locuinţe.

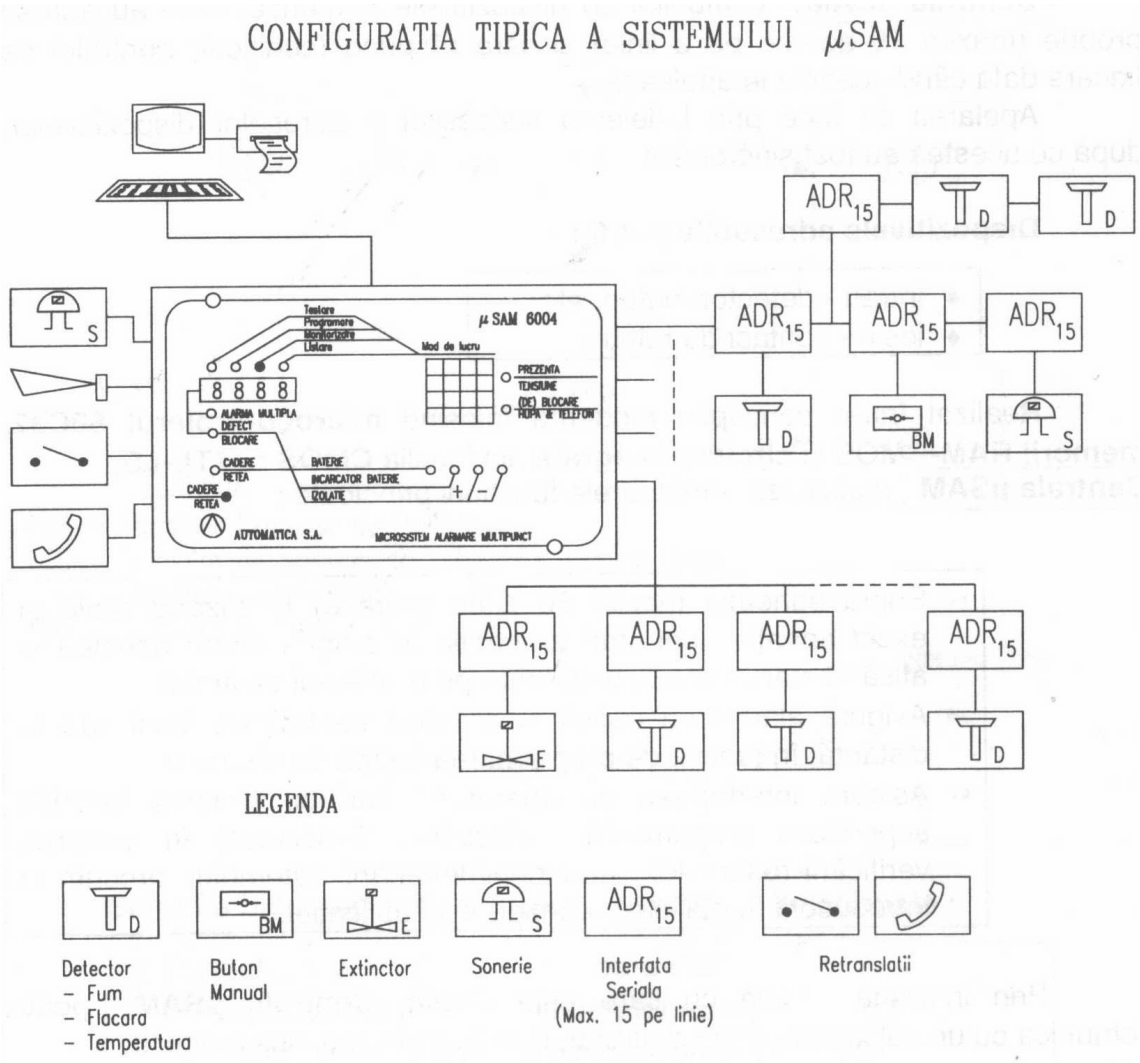


Fig. 5.18 Schema bloc a centralei Micro - SAM

Sistemul din Fig. 5.18 este compus dintr-o centrală de semnalizare şi dispozitive periferice adresabile (butoane, detectoare de fum, de flacără, de temperatură, sonerii, etc.). Centrala permite realizarea a 4 linii (circuite) de semnalizare şi pe fiecare dintre acestea se pot stabili 15 adrese distincte. Fiecare circuit de semnalizare permite conectarea atât a elementelor de iniţiere a alarmei cât şi a elementelor de execuţie (avertizare).

Indicarea evenimentelor care au loc în sistem se efectuează local pe un afişor cu patru caractere sintetizate prin şapte segmente, aparţinând centralei, sau prin tipărire la o imprimantă conectată pe ieşirea serială RS-232 a centralei. Indicarea evenimentelor produse în sistemul de supraveghere se poate efectua şi la distanță, datorită comunicației care se poate stabili între centrala MicroSAM şi un calculator la nivelul ierarhic superior. De asemenea, la distanță se poate comunica automat prin linie telefonică, folosind principiul comutării circuitelor, dispozitivul de apel telefonic având o capacitate de maxim patru numere (abonați).

Centrala MicroSAM transmite interfețelor adresabile mai multe comenzi: SINCRO, APEL, CITIRE si COMANDA. Primele doua comenzi utilizează modularea duratei de suprapunere peste tensiunea continua de alimentare a unei purtătoare de frecventa fixa generata de centrala. Ultimele doua semnale permit citirea informației de la interfață, respectiv transmiterea unei comenzi (active) către aceasta, ambele utilizând variația duratei lipsei purtătoarei susmenționate. Ciclul de funcționare debutează cu alimentarea interfețelor, urmat de transmiterea de către MicroSAM a unui semnal de sincronizare SINCRO.

Fiecare interfața care recepționează acest semnal, va încărca in comparatorul de adrese, adresa proprie, prestabilita, a acesteia. Recepționarea succesiva in continuare a mesajelor APEL de către toate interfețele conectate pe linie va determina ca un singur comparator de adrese sa dea un semnal de egalitate si in acest mod, interfața va putea fi CITITA de către Centrala MicroSAM. Interfața răspunde printr-un impuls de curent de durata variabila, dependenta de starea sa. Centrala interpretează acest semnal si decide daca se reia ciclul cu un alt semnal APEL transmis pe linie sau trimite comanda de acționare a ieşirii.

In starea de veghe a detectorului clasic, conectat la interfață aceasta răspunde printr-un puls de curent de durata corespunzătoare absorbit de detector si rezistenta cap de linie a lui. In starea de alarmare a detectorului, curentul acestuia va creste, crescând si durata răspunsului către MicroSAM.

Verificarea corectitudinii declanşării detectorului se realizează de către centrala printr-o COMANDA, care determina interfața sa întrerupă alimentarea detectorului. Nedeclanşarea imediata a acestuia, permite centralei sa rejecteze alarmele false. De asemenea, netransmiterea pulsului de curent la momentul interogării interfeței indica centralei o defecțiune apărută in aceasta.

Dispozitivele de ieşire sunt in principiu rele bistabile care primesc comenzi de RESET la fiecare CITIRE si SET la fiecare COMANDA.

Sistemul ierarhizat pentru detectare şi alarmare la incendii a fost instalat, pus în funcțiune şi experimentat la laboratorul de automatizări al Facultății de Instalații din Bucureşti. S-a pornit de la echipamentele specializate aflate în producția curentă la AUTOMATICA SA din Bucureşti în anul 1997 şi anume: centrala de semnalizare şi alarmare MicroSAM, un detector de fum, un detector de flacără, un detector de temperatură ridicată şi o interfață specializată pentru conectarea detectoarelor de incendiu la centrală.

Centrala de semnalizare a fost conectată în vederea comunicației cu un calculator PC AT 486 HP Vectra, aflat la nivelul ierarhic superior. Comunicația între centrală şi calculator se efectuează serial, cu o viteză de 2400 bauds, este de tip asincron şi respectă standardul RS-232C.

Sensul transmisiei informației este de la centrala MicroSAM, aflată la nivelul automatizare al sistemului ierarhizat, către calculatorul situat la nivelul supraveghere/gestiune. Este posibilă transmisia informației şi în sens invers, atunci când centrala se programează de la nivelul calculatorului.

Distanța maximă dintre centrala MicroSAM şi calculator poate fi de 15 m, atunci când comunicația se realizează conform standardului RS-232C. Dacă situația concretă dintr-o clădire impune comunicația la distanțe mai mari, atunci se poate utiliza cea de-a doua ieşire serială a centralei, care foloseşte standardul RS-485. La capătul dinspre calculator al cablului de comunicație se va utiliza un adaptor RS-232 / RS-485 şi distanța maximă poate fi de 1200 m.

Caracterele transmise de centrală şi recepționate de calculator (Fig. 5.19) se obțin prin adăugarea la cei opt biți informaționali (codul ASCII al caracterului transmis) a informației cadru, constituită dintr-un bit de START şi un bit de STOP. Sincronizarea emițătorului cu receptorul are loc numai pe durata transmisiei unui caracter.

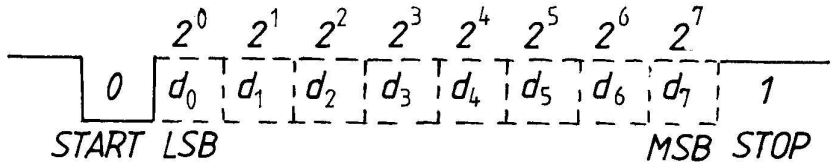


Fig. 5.19 Structura unui caracter transmis de la centrală către calculator.

Aceeaşi ieşire serială a centralei de semnalizare poate fi conectată şi la o imprimantă, unde se tipăresc rapoarte despre starea sistemului de supraveghere şi alarmare la incendii, în momentele semnificative de timp.

În regim de monitorizare, centrala MMicroSAM transmite la calculator sau tipăreşte la imprimantă orice eveniment care se produce în sistem: alarme provenite de la butoane sau detectoare, întreruperi sau scurtcircuite în liniile de semnalizare cablate în clădirile sau spațiile supravegheate, întreruperea alimentării centralei de la rețeaua de 220V c.a., descărcarea acumulatoarelor, etc.

Sistemul ierarhizat pentru detectare şi alarmare la incendii realizat cu MicroSAM (Fig. 5.20) a fost implementat **a sistemului pentru conducerea ierarhizată a instalațiilor**

din clădiri. Calculatorul de la nivelul supraveghere/gestiune este acelaşi calculator care asigură nivelul ierarhic superior în sistemul de conducere al instalației de încălzire şi în sistemul pentru asigurarea şi monitorizarea altor servicii din clădire.

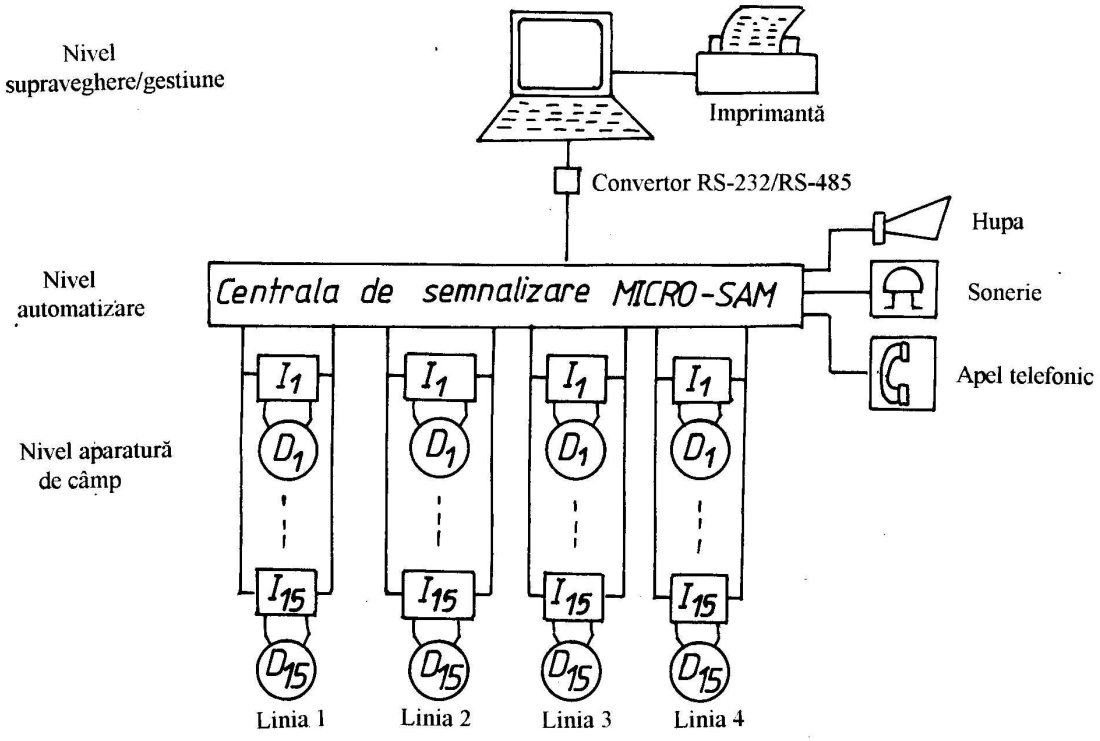


Fig. 5.20 Structura sistemului ierarhizat realizat cu centrala MicroSAM.

În Fig. 5.20 s-au folosit următoarele notații:

- I_1, I_2,...,I_{15} interfețele adresabile ale detectoarelor de incendiu;
- D_1, D_2,...,D_{15} detectoarele de incendiu.

Pentru a putea recepționa mesajele transmise de către centrala de semnalizare, calculatorul de la nivelul supraveghere/gestiune utilizează sistemul de programe KitSAS, completat cu un driver specializat pentru comunicație cu MicroSAM, conceput în limbajul de programare Turbo Pascal 6.0.

Dacă dimensiunile sau alte particularități ale clădirilor supravegheate la incendii, impun utilizarea a mai mult de patru linii de semnalizare cu un total de butoane şi detectoare mai mare de 60 elemente, atunci se poate opta pentru soluții cu mai multe centrale conectate în rețea. Se accentuează în acest mod caracterul distribuit al sistemului de supraveghere şi alarmare la incendii. Centrala MicroSAM nu permite acest mod de funcționare.

6. Bibliografie

1. Asociația inginerilor instalatori din România**, Manualul inginerului instalator**, Artecno, Bucureşti, 2002
2. Astrom K. J**., Control System Design**, Lund Institute of Technology, 2001.
3. Cannon H., R., **Dynamics of physical systems**, McGraw-Hill, N.Z., 1967
4. Chapurlat V., Prunet F., **Presentation du GRAFCET**: GRAphe Fonctionnel de Commande Etape – Transition, www.eerie.fr/~chapurla.
5. Dorf R., C., Bishop R., H., **Modern Control Systems**, Addison-Wesley, New York, 1998.
6. Dumitrache I. Dumitriu S., Mihu I., Munteanu F., Musca Gh., Calcev C., **Automatizări electronice**, Ed. didactică şi pedagogică, Buc., 1993.
7. Dutton K., Thompson S., Barraclough., **The art of control engineering**, Addison-Wesley, New York, 1997.
8. Goodwin G. C., Graebe S. F., Salgado M. E., **Control System Design,** Prentice Hall, N.Y., 2000.
9. Horan T. J., **Control Systems and Applications for HVAC/R**, Prentice Hall Careeer&Technology, N.Y., 1997
10. Ionescu C., Ionescu D., Larionescu S., Bădescu Gr., Ivan N., **Automatizarea instalațiilor.** Îndrumător de proiectare, Vol 1 şi 2, ICB, 1982.
11. Ionescu C., Larionescu S., **Automatizări. Sisteme Automate Discrete Logice**, UTCB, 1997.
12. Ionescu C., Larionescu S., Caluianu S., Popescu D., **Automatizarea Instalațiilor. Comenzi automate**, Matrix Rom, Bucureşti 2002.
13. Ionescu C., Larionescu S., **Decizii prin simulare în rețelele Petri temporizate**, Simpozionul Ştiința modernă şi energia, Universitatea tehnică Cluj Napoca, 1992.
14. Ionescu C., Larionescu S., Ghiauş C., **Implementarea pe automate programabile a rețelelor Petri sigure**, A XXIV-a Conferință de Instalații, Sinaia, 1990.
15. Ionescu C., Larionescu S., **Rețele Petri şi Grafcet**, Conferința a VII, Facultatea de Instalații, Buc., 2001
16. Ionescu C., Larionescu S., **Simularea sistemelor discrete logice cu evoluție paralelă**, A XXV-a Conferință de Instalații, Sinaia, 1991.
17. Ionescu C., Vlădeanu V., Larionescu S., Ionescu D., **Automatizări**, Ed. didactică şi pedagogică, Bucureşti, 1982.
18. Ionescu C.,Larionescu S., **Întocmirea caietelor de sarcini pentru instalațiile de automatizare prin GRAFCET**, Construcții, nr. 9, 1984.
19. Ionescu G., Ionescu V., **Automatica de la A la Z**, Ed. ştiințifică şi enciclopedică, Bucureşti, 1987.

20. Jack H., **Automating Manufacturing System**, http://claymore.engineer.gvsu.edu/, 2005
21. Jack H., **Dynamic System Modeling and Control**, 2001.
22. Jack H., **Manufacturing Controls**, 2002.
23. John K. H., IEC61131-3: **Programming Industrial Automation Systems**, Springer, Berlin, 2001
24. Kirrmann H., **Industrial Automation**, ABB Research Center, Baden, Switzerland, 2005
25. Kwakernaok H., Meinsma G., **Design Methods for Control Systems**, Dutch Institute of Systems and Control, 2001.
26. Landau I., D., **Identificarea şi comanda sistemelor**, Ed. tehnică, Bucureşti, 1997.
27. Larionescu S., **Accente noi în analiza şi proiectarea sistemelor automate**, A XXXVI-a Conferinţă naţională de instalaţii, Sinaia, 2-5 oct.2001, Vol. 2,p.57-65
28. Larionescu S., **Aprecierea robusteţii sistemelor automate**, Măsurări şi Automatizări, Nr. 2, 2001, p.55-56.
29. Larionescu S., **Aspecte moderne în proiectarea inginerească a sistemelor automate**, A XXXVII-a Conferinţă naţională de instalaţii, Sinaia, 1-4 oct., 2002, p.5-18.
30. Larionescu S., **Compararea grafcetului cu reţeaua Petri**, Măsurări şi Automatizări, nr. 3, 2001
31. Larionescu S., **Control Kit**, www.geocities.com/larionescu, 2005.
32. Larionescu S., Ionescu C., Implementarea caietului de sarcini GRAFCET cu automate programabile, Simpozion SIEAR, Bucureşti, 18 noiembrie 1994.
33. Larionescu S., **KitSAS – Simularea şi Analiza Sistemelor**, Matrix Rom, Buc., 1999
34. Larionescu S., **KitSAS** v. 6.12, http://www.geocities.com/larionescu/, 2005
35. Larionescu S., **Proiectarea automatelor cu contacte şi relee orin metoda automatului elementar**, Sesiunea ştiinţifică instalaţii montaj, Bucureşti, 25-26 mai, 1984.
36. Larionescu S., **Schema bloc a unui sistem automat modern**, A XXXV-a Conferinţa de instalaţii, Sinaia, 3-6 oct. 2000, Vol. 2, p. 78-83.
37. Larionescu S., **Sinteza sistemelor discrete logice cu evoluţie paralelă**, A XXV-a Conferinţă de Instalaţii, Sinaia, 1992.
38. Larionescu S., **Teoria sistemelor**, Matrix Rom, Bucureşti, 2006.
39. Larionescu S., **Utilizarea automatelor programabile în gestiunea tehnică a clădirilor**, Simpozionul Facultăţii de Instalaţii, Bucureşti, 1994.
40. Lelic M., **PID Controllers in Nineties**, Science and Technology Division, Corning, Ny, 1999.
41. Leţia T., S., Aştilean A., M., **Sisteme cu evenimente discrete**: modelare, analiză, sinteză şi control, Ed. Albastră, 1998.
42. Licker P.S., **Fundamentals of System analysis**. Boyd / Fraser Publishing Company, Boston, 1987.

43. Mateescu L., **Sisteme informationale si management integrat pentru hoteluri**, www.roel.ro, 2003.
44. Moeller, **Wiring Manual – Automation and Power distribution,** http://www.moeller.ro/suport_documentatie.html, 2006
45. Ould-Bouamama Belkacem, **Regulation automatique**, EUDIL,1998.
46. Păstrăvanu O., **Sisteme cu evenimente discrete**. Tehnici calitative bazate pe formalismul rețelelor Petri, Ed. Matrix Rom, Buc., 1997.
47. Popescu D., **Automate programabile**, Matrix Rom, Bucureşti, 2005.
48. Popescu D., **Teoria sistemelor automate**, Matrix ROM, Buc., 2000.
49. Rivera D. E., **Internal Model Control: A Comprehensive View**, Arizona State University, October, 1999,
50. Riviera D. E., **Introduction to System Identification**, 1998
51. Sârbu L. O., **Contribuţii la studiul şi realizarea unui echipament de supraveghere şi comandă în caz de pericol în clădiri**, Teză de doctorat, Universitatea Tehnică de Construcţii Bucureşti, 2002.
52. Sârbu L., **Micro – SAM, Centrală adresabilă antiincendiu**, Electricianul, Nr. 2, 1997, p. 16 – 18.
53. Scilab Group, **Introduction to Scilab**, INRIA, Unite de recherche de Rocquencourt, Projjet Meta2, France, http://www-rocq.inria.fr/scilab/.
54. Skogestad S., **Probably the best simple PID rules in the world**, AIChE Annual meeting, Reno, NV, USA, Nov. 2001, 28 p.
55. Stătescu T., **Standardizarea sistemelor de alarmă**, Electricianul, Nr. 1, 1997, p. 18 – 19.
56. Taconet B., Chollot B., **Programmation du Grafcet sur automate a langage logique, a relais ou booleen**, Automatisme, feb., 1979.
57. Tham M., T., **Optimality of Control**, University of Newcastle upon tyne, 2002
58. Thamm M. T., **Dynamic Models for Controller Design**, University of Newcastle upon Tyne, 1999.
59. Toulotte J. M., **Reseaux de Petri et automate programmables**, Automatisme, aug. 1978.
60. Traister J. E , **Design & application of security / fire - alarm systems**, McGraw-Hill Book Company, New York, 1981.
61. Willis M. J., Tham M. T., **Advanced Process Control**, University of Newcastle upon Tyne, 2001.